

***Títol:***            **Manteniment remot de laboratoris  
d'informàtica**

***Volum:***           **1 de 1**

***Alumne:***           **Raimon Prats Farré**

***Director/Ponent:*** **Roc Messeguer Pallarès**

***Departament:***   **Arquitectura de Computadors**

***Data:***             **26 de juny de 2007**







---

## DADES DEL PROJECTE

*Títol del Projecte:* **Manteniment remot de laboratoris d'informàtica**

*Nom de l'estudiant:* **Raimon Prats Farré**

*Titulació:* **Enginyeria Tècnica en Informàtica de Sistemes**

*Crèdits:* **15**

*Director/Ponent:* **Roc Messeguer Pallarès**

*Departament:* **Arquitectura de Computadors**

---

## MEMBRES DEL TRIBUNAL *(nom i signatura)*

*President:* **Leandro Navarro Moldes**

*Vocal:* **Glyn Morril**

*Secretari:* **Roc Messeguer Pallarès**

---

## QUALIFICACIÓ

*Qualificació numèrica:*

*Qualificació descriptiva:*

*Data:*

---









## Agraïments

M'agradaria dedicar aquesta pàgina, que és l'última que escric, a totes aquelles persones que m'han recolzat i ajudat, no només durant aquest projecte, sinó durant tots aquests últims anys.

I més concretament, agraeixo als meus companys de feina la paciència que han tingut durant tot aquest temps que, primer deia que faria el projecte, i després, quan el feia, no sabien si estava al despatx perquè em tocava treballar o perquè estava amb el PFC. De tota la gent de la feina, vull dedicar un agraïment especial als meus companys d'Operació i al *Kako*, per haver viscut el tema més d'a prop; al Marc Folch per totes les explicacions informàtiques que m'ha fet; i a l'Ivan Ordoñez per haver-me ajudat en el tema del Rembo. Tampoc m'oblido del centre de reprografia, que ha hagut de tancar més tard perquè pogués acabar de fer les impressions d'aquestes pàgines.

També m'agradaria enviar un agraïment simpàtic i efusiu als meus companys de carrera, que alguns ja fa molts anys que han acabat, perquè sempre m'han ajudat, m'han apreciat i m'han animat! I als meus companys de l'ETSECCPB, inclòs el JL que sempre han confiat en mi. I al Xavi Fuentes.

Tampoc m'oblido de la colla de Capafonts, un lloc encantador que et recupera de qualsevol estrès; ni de la colla del *Ricky*, que els vaig veure començar i acabar les seves carreres i que, per fi, veuran com jo acabo la meua.

I què us en puc dir del meu gran amic Leonadro i de tota la gent que m'ha permès conèixer, inclòs el Mon. Moltes gràcies per haver-me ofert repetidament ajuda i que jo sempre he refusat mesquinament.

Bé, un agraïment general a tots els meus amics i parents que, tot i que no m'han ajudat directament en el projecte, m'han fet passar molt bones estones i m'han fet sentir tan humà.

Ara, en especial, vull agrair al Roc tota la la ajuda prestada durant la lenta evolució d'aquest treball, especialment en els últims moments de més estrès.

I per acabar, un molt especial, sincer i immens agraïment a la Maria José i família, als meus pares i a la meua germana que han fet possible que m'acabi traient la carrera i han conduït la meua vida pel bon camí. GRÀCIES DE TOT COR!

Raimon



# Índex

1. Introducció.....	15
1.1. Presentació.....	15
1.2. Entorn.....	15
1.3. Motivació del projecte.....	16
1.4. Objectius del projecte.....	17
1.5. Diagrama del funcionament general del projecte.....	18
1.6. Planificació del projecte.....	20
2. Descripció de l'entorn del sistema.....	21
2.1. Introducció.....	21
2.2. Tecnologies.....	22
2.2.1. Aplicació web.....	22
2.2.1.1. Arquitectura client-servidor.....	22
2.2.1.2. Servidor web.....	22
2.2.1.3. Apache.....	24
2.2.1.4. PHP.....	25
2.2.1.5. HTML i CSS.....	27
HTML (Hypertext Markup Language).....	27
CSS (Cascading Style Sheets).....	28
2.2.1.6. JavaScript.....	29
2.2.2. SGBD: Sistema de Gestió de Bases de Dades.....	29
2.2.2.1. MySQL.....	32
2.2.3. Arrencada remota.....	33
2.2.3.1. Wake-on-LAN.....	33
2.2.3.2. BOOTP (Bootstrap Protocol).....	35
2.2.3.3. DHCP (Dynamic Host Configuration Protocol).....	35
2.2.3.4. TFTP (Trivial File Transfer Protocol).....	36
2.2.3.5. PXE.....	38
2.2.4. Tecnologia d'imatges i servidor d'imatges.....	40
2.2.4.1. Introducció al concepte d'imatges de software.....	40
2.2.4.2. Servidor d'imatges.....	41
2.2.4.3. Rembo.....	42
2.3. Treballs relacionats.....	43
3. Anàlisi de requisits.....	45
3.1. Introducció.....	45
3.2. Requisits funcionals.....	46
3.2.1. Validació.....	46
3.2.1.1. Usuaris no validats.....	47
3.2.1.2. Usuaris validats.....	47
3.2.2. Consulta a l'inventari (de PCs, laboratoris i edificis).....	48
3.2.3. Gestió d'estats (de PCs i laboratoris).....	48
3.2.4. Gestió d'imatges.....	49
3.2.5. Manteniment (de PCs i laboratoris).....	49
3.3. Requisits no funcionals.....	50

3.3.1. Interfície.....	51
3.3.2. Qualitat del software.....	51
3.3.3. Càrrega.....	52
3.3.4. Tecnològics.....	53
3.3.4.1. Comprovació del hardware disponible.....	54
4. Especificació i disseny.....	55
4.1. Especificació.....	56
4.1.1. Casos d'ús.....	56
4.1.1.1. Actors.....	56
4.1.1.2. Diagrames dels casos d'ús.....	57
Casos d'ús de validació.....	57
Casos d'ús de consulta a l'inventari (de PCs, laboratoris i edificis).....	58
Casos d'ús de gestió d'estats (de PCs i laboratoris).....	58
Casos d'ús de gestió d'imatges.....	59
Casos d'ús de manteniment (de PCs i laboratoris).....	59
4.1.1.3. Especificació dels casos d'ús.....	60
Casos d'ús de validació.....	60
Casos d'ús de consulta a l'inventari (de PCs, laboratoris i edificis).....	61
Casos d'ús de gestió d'estats (de PCs i laboratoris).....	63
Casos d'ús de gestió d'imatges.....	66
Casos d'ús de manteniment (de PCs i laboratoris).....	68
4.2. Disseny.....	71
4.2.1. Diagrama de classes.....	72
4.2.2. Disseny lògic de la base de dades.....	73
4.2.3. Diagrames d'activitats.....	73
4.2.3.1. Diagrama d'activitats per a usuaris no validats.....	75
4.2.3.2. Diagrama d'activitats per a usuaris ja validats.....	76
4.2.4. Arquitectura del software.....	77
4.2.4.1. Patrons arquitectònics i de disseny.....	77
4.2.4.2. El patró Model-Vista-Controlador i la seva adaptació per a web.....	78
4.2.4.3. Controlador façana.....	79
4.2.5. Diagrama de classes amb el patró MVC.....	80
4.2.6. Diagrames de seqüència dels casos d'ús.....	82
4.2.6.1. Mostra edificis.....	82
4.2.6.2. Mostra laboratoris.....	83
4.2.6.3. Mostra PCs.....	83
4.2.6.4. Mostra PC.....	84
4.2.6.5. Canvia estat laboratoris.....	84
4.2.6.6. Canvia estat PCs.....	85
4.2.6.7. Canvia estat PC.....	85
4.2.6.8. Mostra historial estat laboratoris.....	86
4.2.6.9. Mostra historial estat PCs.....	86
4.2.6.10. Mostra historial estat PC.....	87
4.2.6.11. Crea imatge.....	87
4.2.6.12. Modifica imatge.....	88
4.2.6.13. Esborra imatge.....	88
4.2.6.14. Restaura laboratoris.....	89
4.2.6.15. Restaura PCs.....	90

4.2.6.16. Restaura PC.....	91
5. Implementació.....	93
5.1. Software usat.....	93
5.1.1. Notes sobre PHP.....	94
5.2. Aspectes de seguretat.....	95
5.2.1. Sistema d'autenticació.....	95
5.3. Disseny de la interfície amb l'usuari.....	95
5.4. Documentació.....	98
5.5. Fitxers de l'aplicació.....	98
5.6. Base de dades.....	100
5.6.1. Creació de la base de dades.....	100
5.6.2. Disseny físic de la base de dades.....	100
5.6.3. Primeres insercions.....	103
5.7. Exemples de codi.....	103
5.7.1. index.html: el controlador façana.....	103
5.8. Proves.....	105
6. Conclusions.....	107
6.1. Balanç.....	107
6.2. Línies de futur.....	108
7. Bibliografia.....	109



# 1. Introducció

1.1. Presentació.....	15
1.2. Entorn.....	15
1.3. Motivació del projecte.....	16
1.4. Objectius del projecte.....	17
1.5. Diagrama del funcionament general del projecte.....	18
1.6. Planificació del projecte.....	19

## **1.1. Presentació**

El que farem en aquest projecte serà dissenyar i desenvolupar un sistema prototipus capaç de restaurar remotament (a distància) les imatges de PCs de diversos laboratoris informàtics, tant interactivament, com amb el moment d'inici programat amb anterioritat. Per fer-ho, dissenyarem i implementarem una aplicació accessible via web (amb autenticació requerida), que ens permetrà dur a terme les restauracions. També reportarem com han anat aquestes restauracions i emmagatzemarem els reports en un servidor. Aquestes restauracions es podran dur a terme encara que els PCs estiguin apagats abans de començar. A més, emmagatzemarem en un inventari informació útil dels PCs, dels laboratoris, dels edificis i de les imatges.

## **1.2. Entorn**

En el moment de fer aquest projecte, ja fa anys que estem treballant com a operadors informàtics al CBL (Campus del Baix Llobregat) de la UPC. Aquest campus és situat a Castelldefels i està integrat dins del PMT (Parc Mediterrani de la Tecnologia).

Actualment, en el campus de l'UPC hi ha diversos edificis amb laboratoris informàtics:

## 1. Introducció

---

- Edifici C4, on hi ha l'EPSC (Escola Politècnica Superior de Castelldefels). Té 13 laboratoris informàtics.
- Edifici D4, on hi ha l'ESAB (Escola Superior d'Agricultura de Barcelona). Té 4 laboratoris informàtics.
- Edifici D7, de Serveis. Aquí, a més d'altres serveis, hi ha la Biblioteca del Campus. Té 1 laboratori informàtic.
- Altres edificis del PMT. Com que no són de l'UPC, no els tindrem en compte.

Actualment, per fer el manteniment dels laboratoris esmentats, s'usa un sistema d'imatges que s'ha implementat *ad hoc* basant-se en el programa Rembo versió 4.0. Un servidor conté totes les imatges dels PCs del Campus. Tots els PCs del campus tenen activada l'opció d'arrencar de xarxa. Llavors, quan un usuari encén un PC, abans d'iniciar-se el sistema operatiu, l'ordinador consulta la xarxa i, a través d'aquesta, se li envia un menú que dona l'opció a l'usuari d'iniciar algun sistema operatiu, o bé restaurar alguna imatge del PC.

Part de la nostra feina habitual consisteix a realitzar revisions periòdiques *in situ* dels diversos laboratoris informàtics del campus. Aquestes revisions, consisteixen, essencialment, a:

1. Comprovar el *hardware* dels PC (i reparar, també *in situ*, els que es puguin amb els mitjans que tenim a mà).
2. Restaurar-ne el *software* usant la tecnologia de les imatges anteriorment esmentada.
3. Retirar els PC que no s'han pogut reparar al moment.
4. Reportar les anomalies del laboratori.

Tot i que no forma part de les nostres responsabilitats dissenyar la rutina de revisions, a còpia de fer-ne, ens hem adonat que és una feina molt repetitiva i automatitzable. A més, com que el CBL té molts laboratoris, amb molts PCs cadascun (de 12 a 24), aquestes tasques repetitives es multipliquen. Això també comporta haver-se de desplaçar físicament amunt i avall.

### **1.3. Motivació del projecte**

La situació exposada al punt anterior, ens va conduir a idear possibles mesures per millorar i optimitzar el sistema de manteniment de laboratoris: la primera que se'ns va acudir, va ser no haver de restaurar el *software* dels PCs un per un, sinó poder-los fer tots de cop amb una sola ordre de restauració.



Després, també se'ns va ocórrer que, per evitar desplaçaments, una part del manteniment es pogués fer de forma remota. Això comportava crear una eina perquè l'operador de laboratori pogués escollir de quin laboratori volia fer la revisió.

A partir d'aquestes idees inicials, de seguida vam veure que necessitàvem tot d'elements afegits perquè es pogués portar a bon terme. Per exemple, per fer aquesta idea factible necessitaríem una base de dades amb tots els laboratoris del campus, amb tots els PC del campus, amb totes les imatges que s'usen per restaurar el *software*, etc. També necessitaríem un servidor que ens proporcionés l'eina de restauració de laboratoris de manera remota. Al CBL, ja n'hi havia un (de servidor d'imatges) i potser el podríem usar.

I no només vam veure que necessitàvem eines, sinó que se'ns va obrir un ventall de funcionalitats noves i que ens semblaven molt útils. Per exemple, podríem programar l'inici d'aquestes restauracions perquè comencessin a fer-se en una hora en què no hi hagués docència (com a la matinada), i així es podrien dur a terme sense haver de fer fora cap possible usuari.

Arribats en aquest punt, ja teníem clar que aquestes idees podrien donar lloc a un projecte de final de carrera de la FIB.

### **1.4. Objectius del projecte**

Com ja hem dit anteriorment, l'objectiu principal del projecte és dissenyar i desenvolupar un sistema prototipus capaç de restaurar remotament (a distància) les imatges dels PCs dels diversos laboratoris del CBL (tant interactivament, com amb el moment d'inici programat amb anterioritat), usant una aplicació accessible via web (amb autenticació requerida), i reportar (i emmagatzemar en un servidor) com han anat aquestes restauracions. A més, aquest sistema ha de poder executar aquestes restauracions encara que els PCs estiguin apagats abans de començar. També ha de poder emmagatzemar en un inventari informació útil dels PCs, dels laboratoris, dels edificis i de les imatges.

Concretem els objectius en els ítems següents:

1. Dissenyar i implementar una base de dades relacional que centralitzi la informació de l'inventari de PCs, laboratoris, edificis, imatges, i més informació útil.
2. Programar una aplicació accessible via web (http o https) que permeti dur a terme les següents accions:
  - ◇ Validar-se per impedir que qualsevol persona NO autoritzada usi el projecte.

## 1. Introducció

---

- ◇ Simular la restauració de les imatges d'un o més PCs o laboratoris de manera interactiva. També, simular la restauració podent programar el moment d'inici amb anterioritat i permetre repeticions periòdiques i automàtiques d'aquestes restauracions.
  - ◇ Mostrar les dades de l'inventari de PCs, laboratoris i edificis emmagatzemades a la base de dades del primer ítem. Aquestes dades, només es consultaran i es mostraran.
  - ◇ Gestionar (mostrar i permetre modificar) les dades de les imatges dels PCs del CBL, i també assignar aquestes imatges a un o més PCs o laboratoris.
  - ◇ Gestionar (mostrar i permetre modificar) l'estat dels PCs del CBL (si estan operatius o bé fora de servei), i l'estat dels laboratoris del CBL (si estan oberts o bé tancats).
3. Engegar els PCs a distància.
  4. Adaptar del sistema d'imatges dels PCs dels laboratoris del CBL perquè s'ajusti a les funcions del projecte; o bé, fer-ne una còpia i treballar amb aquesta; o bé, crear-ne un de nou i afegir-li només les imatges per fer les proves.
  5. Reportar l'estat de les restauracions i emmagatzemar-lo (el report) en un servidor.
  6. Provar la viabilitat d'aquest sistema prototipus en un escenari simulat.

### ***1.5. Diagrama del funcionament general del projecte***

Una vegada definits els objectius del projecte, il·lustrem amb un diagrama esquemàtic el seu funcionament general. En aquest esquema hi apareixen alguns conceptes, noms o protocols que encara no hem explicat. Els explicarem al capítol Descripció de l'entorn del Sistema.

## 1.5. Diagrama del funcionament general del projecte

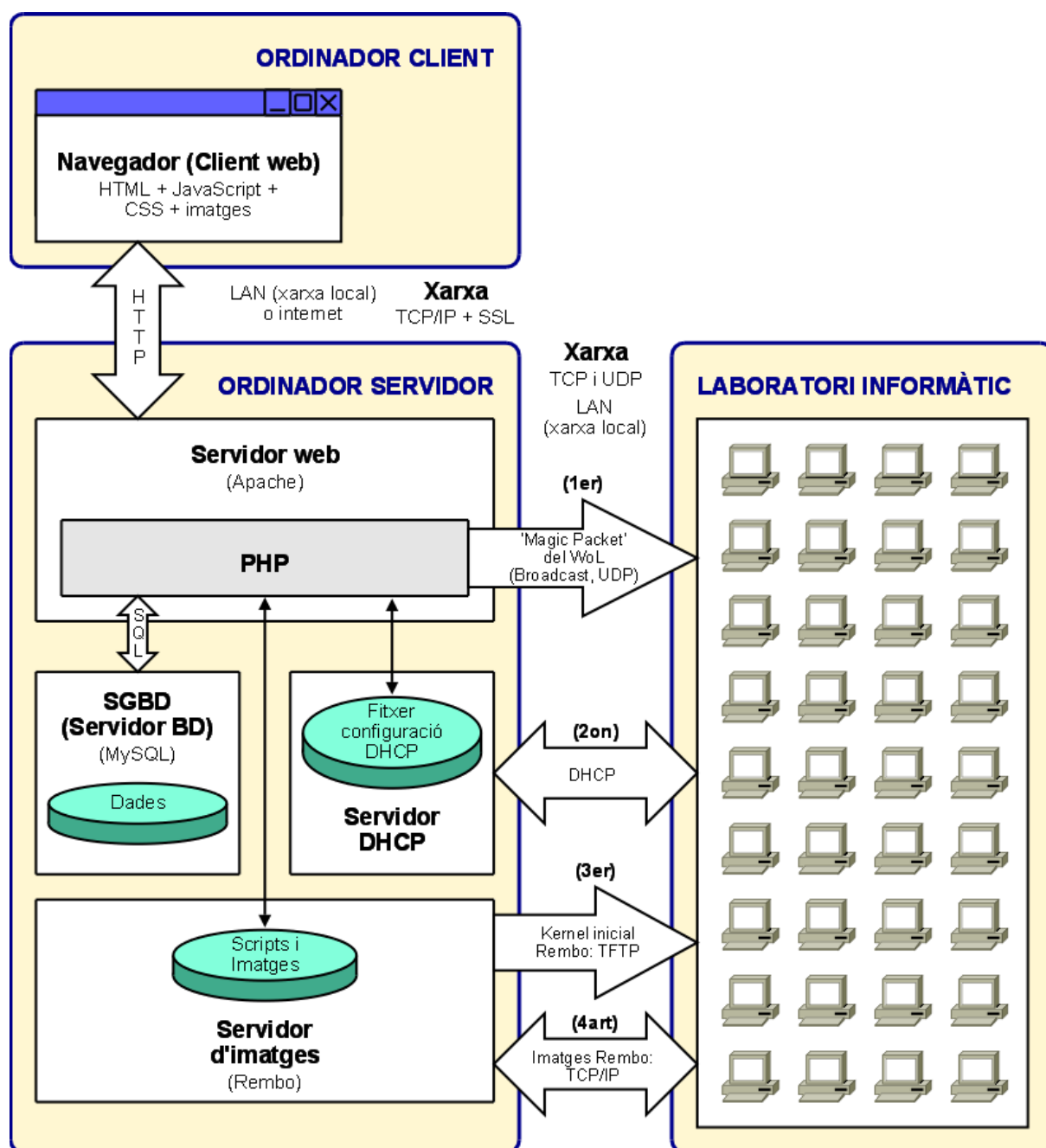


Figura 1.1: Esquema del funcionament general del sistema

### **1.6. Planificació del projecte**

En aquest apartat mirem les tasques en què s'ha dividit el projecte per poder dur a terme els objectius descrits en l'apartat anterior i els assignem temps. Són les següents:

	<b>Temps necessari</b>
● Descripció de l'entorn del sistema	<b>3 setmanes</b>
● Anàlisi de requisits	<b>2 setmanes</b>
● Especificació	<b>2 setmanes</b>
● Disseny	<b>3 setmanes</b>
● Implementació i proves	<b>6 setmanes</b>
● Documentació	<b>4 setmanes</b>

## 2. Descripció de l'entorn del sistema

2.1. Introducció.....	21
2.2. Tecnologies.....	22
2.2.1. Aplicació web.....	22
2.2.1.1. Arquitectura client-servidor.....	22
2.2.1.2. Servidor web.....	22
2.2.1.3. Apache.....	24
2.2.1.4. PHP.....	25
2.2.1.5. HTML i CSS.....	27
HTML (Hypertext Markup Language).....	27
CSS (Cascading Style Sheets).....	28
2.2.1.6. JavaScript.....	29
2.2.2. SGBD: Sistema de Gestió de Bases de Dades.....	29
2.2.2.1. MySQL.....	31
2.2.3. Arrencada remota.....	34
2.2.3.1. Wake-on-LAN.....	34
2.2.3.2. BOOTP (Bootstrap Protocol).....	35
2.2.3.3. DHCP (Dynamic Host Configuration Protocol).....	36
2.2.3.4. TFTP (Trivial File Transfer Protocol).....	37
2.2.3.5. PXE.....	39
2.2.4. Tecnologia d'imatges i servidor d'imatges.....	40
2.2.4.1. Rembo.....	40
2.3. Treballs relacionats.....	41

### 2.1. Introducció

En aquest projecte i la seva documentació usem una sèrie de conceptes i eines tecnològiques, sense formar part dels objectius del projecte però necessaris per poder-lo dur a terme, que anomenem entorn del sistema. Per facilitar la comprensió del projecte, expliquem aquests conceptes a l'apartat de Tecnologies d'aquest capítol.

Després, a l'apartat Treballs Existents, analitzarem altres eines que ja podem trobar al mercat i que fan coses similars a aquest treball. Les compararem i veurem quins pros i contres tenen en relació amb aquest projecte.

### **2.2. Tecnologies**

Els conceptes i eines que explicarem en aquest capítol són: aplicació web, Sistema Gestor de Bases de Dades (SGBD), sistema d'imatges d'ordinadors i alguns protocols de xarxa.

#### **2.2.1. Aplicació web**

Una aplicació web és una aplicació dissenyada per servir pàgines web des d'un servidor web i per ser usada per usuaris a través de clients web.

Perquè aquesta definició quedi més clara, ara explicarem què és un servidor web i també alguns protocols i llenguatges de programació relacionats amb ell.

##### **2.2.1.1. Arquitectura client-servidor**

Per entendre el concepte de servidor, hem d'explicar què és l'arquitectura client-servidor, en la qual hi ha uns clients usuaris d'un sistema que fan peticions a un servidor. Aquest, en rebre les peticions, les processa i en retorna les respostes als clients.

Un client és un *software* que, per fer certes accions, necessita quelcom que no pot obtenir sense fer peticions a un *software* extern. Aquest, és el servidor i la seva funció és esperar peticions de clients i donar-ne les respostes.

##### **2.2.1.2. Servidor web**

Un servidor web és un programa que ofereix pàgines web. Els clients d'aquest són els navegadors dels ordinadors dels usuaris de les pàgines web.

Una aplicació web (servida per un servidor web) és un exemple típic de l'arquitectura client-servidor. Com a clients d'un servidor web, els clients són els navegadors dels ordinadors dels usuaris de l'aplicació. Quan algun usuari introdueix una adreça web al seu navegador, aquest sol·licita la pàgina al servidor, el qual la genera i l'envia al programa client. Aquest client, finalment, mostra la pàgina web rebuda a l'usuari que l'havia demanada.

Els programes que usem en aquesta aplicació web: l'Apache com a servidor web de pàgines estàtiques; amb un mòdul afegit que és el PHP i s'usa per servir pàgines dinàmiques. Aquest servidor web, a més, es comunica amb un Sistema Gestor de Bases de Dades (SGBD) que és el MySQL.

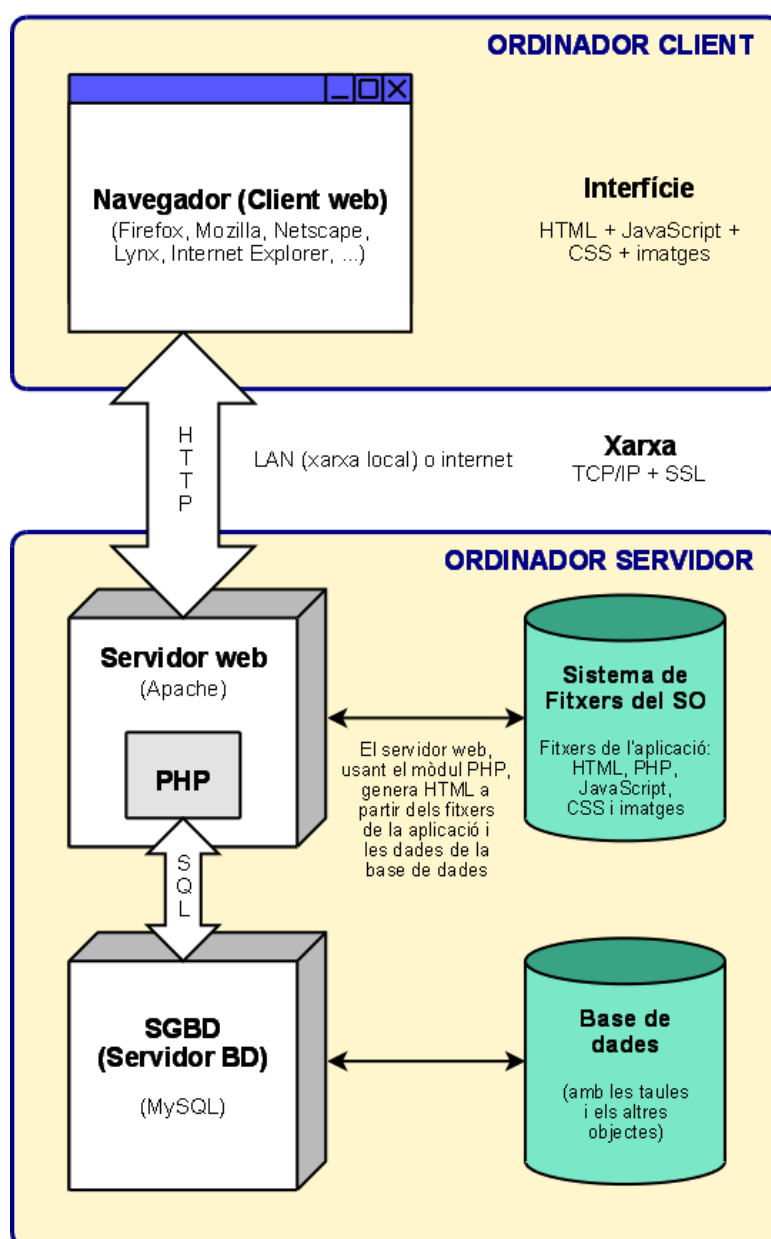


Figura 2.1: Esquema de funcionament d'una aplicació web

### **2.2.1.3. Apache**

L'Apache és un *software* servidor web. És un dels principals triomfadors del programari lliure perquè aproximadament un 63% dels servidors web l'utilitzen.

Les característiques principals que té són les següents:

- Multiplataforma
- Dóna suport a HTTP 1.1
- Modular (adaptable)
- Codi obert
- Extensible (PHP, SQL, Perl)
- Ràpid
- Eficient
- Quota de mercat

El protocol HTTP és el que fan servir els servidors i els clients per comunicar-se. Per tant, és un protocol molt important i usat a internet. A continuació mostrem un exemple de funcionament d'aquest protocol HTTP, amb pàgines dinàmiques i bases de dades:



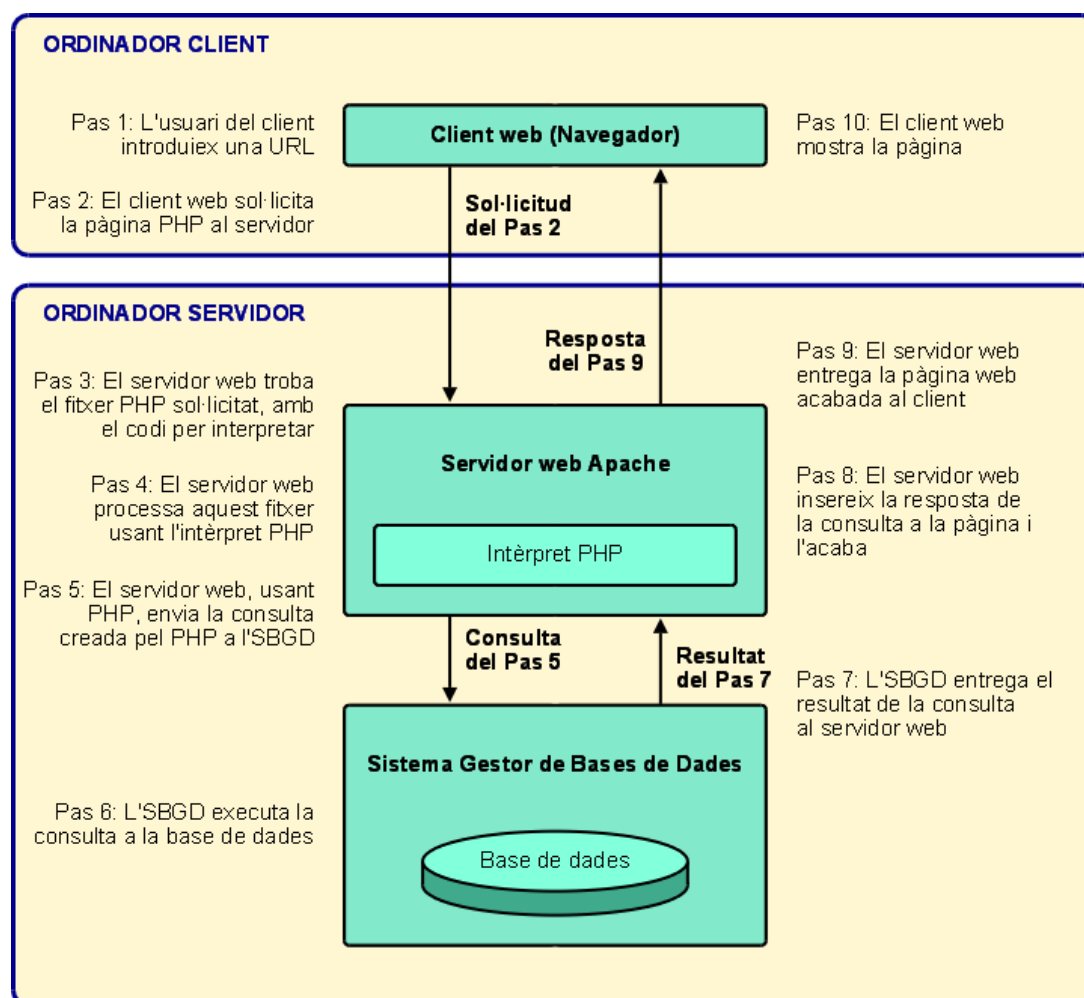


Figura 2.2: Esquema del funcionament d'HTTP amb PHP i bases de dades

Per a més informació sobre el servidor Apache: <http://www.apache.org/>.

#### 2.2.1.4. PHP

El PHP (acrònim recursiu de *PHP: Hypertext Preprocessor* en anglès) és un llenguatge de seqüència de comandes de servidor dissenyat específicament per produir pàgines web dinàmiques. És interpretat en el servidor i genera codi HTML que és l'únic que el client veu (el client NO veu el codi PHP). Dins una pàgina web es pot incrustar codi PHP que s'executarà cada vegada que es visiti aquesta.

## 2. Descripció de l'entorn del sistema

---

Fou concebut l'any 1994. Ha experimentat transformacions importants fins convertir-se en el producte actual, l'última versió és la 5.

És un producte de codi obert, que significa que es pot obtenir el codi, usar-lo, modificar-lo i redistribuir-lo sense cap cost. És del tot gratuït.

La seva sintaxi es basa en la d'altres llenguatges de programació com C i Perl. Així, el procés d'aprenentatge del llenguatge PHP no és complicat si es coneix algun d'aquests llenguatges o d'altres similars com C++ o Java.

Entre els competidors principals de PHP es pot citar a Perl, ASP (Active Server Pages) de Microsoft, JSP (Java Server Pages) i Allaire ColdFusion. En comparació amb aquests altres productes, PHP presenta molts avantatges:

- Alt rendiment
- Interfícies per a una gran quantitat de sistemes de bases de dades diferents (MySQL, PostgreSQL, mSQL, Oracle, dbm, filePro, Hyperwave, Informix, Interbase, Sybase, etc)
- Biblioteques incorporades per a moltes tasques web habituals
- Baix cost, si no nul
- Facilitat d'aprenentatge
- Portabilitat
- Accés al codi obert
- Permet les tècniques d'orientació a objectes

Pel que fa al rendiment, el PHP és molt eficient. Mitjançant l'ús d'un únic servidor web amb PHP, es poden servir milions d'accessos al dia. Així ho reflecteixen els indicadors comparatius de rendiment publicats per Zend Technologies, que mostren com PHP supera àmpliament als seus competidors en aquesta faceta.

Com hem esmentat abans, PHP incorpora una gran quantitat de funcions integrades per realitzar tasques relacionades amb la web. Permet generar imatges GIF a l'instant, establir connexions a altres serveis de xarxa, enviar correus electrònics, generar documents PDF, treballar amb *cookies* i realitzar altres tasques.

Pel que fa a la portabilitat, remarcuem que el codi PHP funciona sense necessitat d'aplicar cap modificació en tots els sistemes operatius gratuïts del tipus Unix (Linux, FreeBSD), versions comercials de Unix (Solaris, IRIX) i en les diferents versions de Microsoft Windows.

Per a més informació sobre el PHP: <http://www.php.net/>.

### 2.2.1.5. HTML i CSS

#### HTML (Hypertext Markup Language)

L'HTML (*Hypertext Markup Language* en anglès) és un llenguatge d'etiquetes dissenyat per estructurar textos i presentar-los en forma d'hipertext, que és el format estàndard de les pàgines web.

S'entén per hipertext un document digital que es pot llegir de manera no seqüencial, o sigui, que enmig del text d'aquest document, hi ha enllaços a altres parts del document a les que s'hi pot anar i després, si es vol, tornar-ne. Gràcies a Internet i als navegadors més populars com Internet Explorer, Mozilla, Firefox o Netscape, l'HTML s'ha convertit en un dels formats més populars que existeixen per a la creació de documents.

L'HTML usa etiquetes o marques, que són breus instruccions d'inici o final. A través d'aquestes, es determina la forma en què ha d'aparèixer al navegador el text, les imatges i la resta d'elements.

Tota etiqueta s'identifica perquè està tancada entre els signes menor que (<) i major que (>). A més, algunes etiquetes tenen atributs que poden prendre algun valor.

No hi ha cap especificació oficial de l'HTML 1.0 perquè ja existien múltiples estàndards informals de l'HTML quan es va decidir crear un estàndard oficial. L'any 1993 es va crear un successor de l'HTML anomenat HTML+. Aquest, va ser dissenyat originalment com un superconjunt de l'HTML que permetés evolucionar gradualment des del format HTML anterior. La primera especificació formal de l'HTML+ va prendre el nom de versió 2.0 per distingir-la d'aquells estàndards no oficials previs. HTML+ mai va arribar a convertir-se en un estàndard.

L'esborrador de l'estàndard HTML 3.0 va ser proposat pel W3C (World Wide Web Consortium) l'any 1995. Es van introduir moltes noves capacitats. Es va dissenyar per ser compatible amb l'HTML 2.0, però era massa complex per ser implementat amb la tecnologia d'aquella època. Així, que es va haver d'abandonar degut a la carència de recolzament per part dels fabricants de navegadors web. L'HTML 3.1 mai va arribar a ser proposat oficialment. El següent, l'HTML 3.2, abandonava la majoria de noves característiques de l'HTML 3.1 i, a canvi, adoptava molts elements desenvolupats inicialment pels navegadors web Netscape i Mosaic.

L'HTML 4.0 també va adoptar molts elements específics desenvolupats inicialment per un navegador web concret, però al mateix temps va començar a netejar l'HTML assenyalant alguns d'ells com 'desaprovat'.

## 2. Descripció de l'entorn del sistema

---

Ja no hi haurà noves versions d'HTML. Tot i així, l'herència de l'HTML es manté en l'XHTML, que es basa en l'XML.

Per a més informació sobre l'HTML: <http://www.w3.org/>.

### **CSS (Cascading Style Sheets)**

Els fulls d'estil en cascada (*Cascading Style Sheets*) són un llenguatge formal usat per a definir la presentació d'un document estructurat escrit en HTML o XML. El W3C (World Wide Web Consortium) és l'encarregat de formular l'especificació dels fulls d'estil que servirà d'estàndard per als agents d'usuari o navegadors.

La idea que hi ha darrera del desenvolupament de CSS és la de separar l'estructura d'un document de la seva presentació.

Els avantatges d'utilitzar CSS són:

- Control centralitzat de la presentació d'una pàgina web, amb el que s'aconsegueix agilitzar de manera considerable l'actualització d'aquesta.
- Els navegadors permeten als usuaris especificar el seu propi full d'estil local que serà aplicat a una pàgina web remota, cosa que fa augmentar considerablement l'accessibilitat. Per exemple, persones amb deficiències visuals poden configurar el seu propi full d'estil per a augmentar la mida del text o remarcar més els links.
- Una pàgina pot disposar de diferents fulls d'estil segons el dispositiu que la mostri o, fins i tot, pot permetre a l'usuari l'elecció de diferents estils. Per exemple, per ser impresa, mostrada en un dispositiu mòbil, o ser 'llegida' per un sintetitzador de veu.
- El document (XML, XHTML o HTML) és més clar d'entendre i s'aconsegueix reduir considerablement la seva mida.

Per a més informació sobre els CSS: <http://www.w3.org/>.

### **2.2.1.6. JavaScript**

JavaScript és un llenguatge interpretat pensat per a pàgines web, per ser inclòs dins de documents HTML permetent afegir interactivitat a les webs. Té una sintaxi semblant a la del llenguatge Java.

Usar JavaScript permet estalviar-se algunes de les tasques que el servidor hauria de fer (a través de PHP en aquest projecte, o bé a través d'ASP, Java o CGI en altres projectes) i per tant, ajuda a no carregar tant ni el servidor ni la xarxa.

El llenguatge fou inventat per Brendan Eich de l'empresa Netscape Communications, qui fabricà els primers navegadors d'Internet comercials.

Tradicionalment, JavaScript es s'usava en pàgines web HTML, per fer tasques i operacions en el marc de l'aplicació client-servidor. Amb la irrupció de Web 2.0, JavaScript s'ha convertit en un veritable llenguatge de programació que utilitza la potència de càlcul del navegador per augmentar la usabilitat d'aplicacions web amb tècniques avançades com Ajax o JCC.

Les característiques més importants que té són les següents:

- No fa comprovacions de tipus de cap classe
- Té expressions, funcions i sentències de control de flux d'execució
- Suporta esdeveniments
- És interpretat pel navegador quan es llegeix a la pàgina. Això implica que el navegador ha de tenir suport per a JavaScript per poder utilitzar-lo. Actualment, tots els navegadors suporten JavaScript.
- És orientat a objectes, però no té els conceptes de classe ni d'herència.

### **2.2.2. SGBD: Sistema de Gestió de Bases de Dades**

Un Sistema de Gestió de Bases de Dades (SGBD o bé DBMS de *DataBase Management System* en anglès) és un conjunt de programes informàtics molt específics dissenyats per facilitar la gestió de dades en una base de dades. Aquesta gestió controla l'organització, l'emmagatzemament i la consulta de dades de la base de dades. Un SGBD pot gestionar diverses bases de dades alhora.

## 2. Descripció de l'entorn del sistema

---

1. **Organització de les dades:** Els SGBD fan servir un llenguatge de modelat per definir l'esquema de cada base de dades hostatjada en ells. Aquest esquema segueix un model de dades de defineix com s'han de representar, organitzar i usar les dades de l'SGBD.

En el nostre cas, les dades s'organitzen seguint el *model relacional*<sup>1</sup>. Per tant, el nostre SGBD és un SGBD relacional.

2. **Emmagatzemament de les dades:** El SGBD emmagatzemen les dades en estructures de dades (camp, registres i fitxers) optimitzades per tractar amb grans quantitats de dades. Aquestes estructures de dades es desen en un dispositiu d'emmagatzemament permanent de dades (molt més lent que la memòria RAM) de manera ordenada i optimitzada.
3. **Consulta i manipulació de les dades:** Els SGBD usen llenguatges de consulta per demanar informació, analitzar i actualitzar les seves bases de dades, d'acord amb els privilegis que tingui cada usuari de les dades.

A més, tot SGBD ha de complir diferent objectius:

- **Abstracció de la informació:** Els SGBD han d'estalviar als usuaris detalls sobre l'emmagatzemament físic de les dades. Ha de ser igual si una base de dades ocupa un o centenars d'arxius. Això ha de ser transparent per a l'usuari. Hi ha diversos nivells d'abstracció.
- **Independència:** La independència de les dades consisteix en la capacitat de modificar l'esquema (físic o lògic) d'una base de dades sense haver de realitzar canvis en les aplicacions que es serveixen d'ella.
- **Redundància mínima:** Un bon disseny d'una base de dades ha d'aconseguir evitar l'aparició d'informació repetida o redundant. D'entrada, l'ideal és aconseguir una redundància nul·la. No obstant això, en alguns casos la complexitat dels càlculs fa necessària l'aparició de redundàncies.
- **Consistència:** En aquells casos en els quals no s'ha aconseguit aquesta redundància nul·la, serà necessari vigilar que aquella informació que apareix repetida s'actualitzi de forma coherent, és a dir, que totes les dades repetides s'actualitzin de forma simultània.
- **Seguretat:** La informació emmagatzemada en una base de dades pot arribar a tenir un gran valor. Els SGBD han de garantir que aquesta informació es troba assegurada davant usuaris malintencionats, que intentin llegir informació privilegiada; davant atacs que desitgin manipular o que destruir la informació; o simplement davant de les bestieses d'algun usuari autoritzat però despistat. Normalment, els SGBD disposen d'un complex sistema de permisos a usuaris i grups d'usuaris, que permeten atorgar diverses categories de permisos.

---

<sup>1</sup>Per a més informació sobre el *model relacional*, podeu consultar els següents enllaços: (1) [A relational Model of Data for Large Shared Data Banks](#), (2) [Teradata Magazine | In the Beginning: An RDBMS history](#), o bé (3) [Open Directory - Computers: Software: Databases: Relational](#).

- **Integritat:** Els SGBD, han d'adoptar les mesures necessàries per garantir la validesa de les dades emmagatzemades. És a dir, protegir les dades davant d'errors de maquinari, dades introduïdes per usuaris descuidats, o qualsevol altra circumstància capaç de corrompre la informació emmagatzemada.
- **Suport i recuperació:** Els SGBD han de proporcionar una forma eficient de realitzar còpies de seguretat de la informació emmagatzemada en ells, i de restaurar a partir d'aquestes còpies les dades que s'hagin pogut perdre.
- **Control de la concurrència:** En la majoria d'entorns (excepte potser el domèstic), el més habitual és que siguin moltes les persones que accedeixen a una base de dades, bé per recuperar informació, o bé per emmagatzemar-la. I és també freqüent que els esmentats accessos es realitzin de forma simultània. Així doncs, un SGBD ha de controlar aquest accés concurrent a la informació, que podria derivar en inconsistències.
- **Temps de resposta:** Lògicament, és desitjable minimitzar el temps que el SGBD tarda a donar-nos la informació sol·licitada i a emmagatzemar els canvis realitzats.

Per acabar, alguns avantatges i inconvenients d'usar un SGBD:

➔ **Avantatges:**

- ✓ Facilitat de manegar grans volums d'informació.
- ✓ Optimització del temps i l'espai.
- ✓ Independència del tractament d'informació.
- ✓ Seguretat de la informació i accés a usuaris autoritzats.
- ✓ No hi ha duplictat d'informació, comprovació d'informació en el moment d'introduir la mateixa.
- ✓ Integritat referencial.

➔ **Inconvenients:**

- x El cost d'actualització del maquinari (si cal) i programari (si és de pagament) són molt elevats.
- x Cost (salari) de l'administrador de la base de dades és costós.
- x El mal ús d'alguna de les capacitats del SGBD pot originar greus problemes.

### 2.2.2.1. MySQL

El MySQL és un SGBD relacionals multi-usuari i de sub-processament múltiple. Usa, com el seu nom indica, SQL (Structured Query Language), el llenguatge estàndard per a la consulta de bases de dades utilitzat arreu del món. Encara que no compleix tots els estàndards SQL més coneguts, ofereix les funcionalitats necessàries per poder desenvolupar el nostre projecte.

El MySQL és un sistema per a l'administració de bases de dades relacionals ràpid i sòlid. Les seves bases de dades ens permeten emmagatzemar, buscar, ordenar i recuperar dades de manera eficient. El servidor de bases de dades de MySQL controla l'accés a les dades per garantir l'ús simultani de diversos usuaris, per proporcionar accessos a aquestes dades i per assegurar-se que només obtenen accés a aquestes dades els usuaris amb autorització.

Actualment, MySQL es distribueix sota una llicència de codi obert, però també existeixen llicències comercials.

Entre els competidors principals de MySQL, podem citar PostgreSQL, Microsoft SQL Server i Oracle.

Dels avantatges que presenta MySQL en destaquem els següents:

- **Alt rendiment.** MySQL és molt ràpid, els indicadors comparatius mostren en molts casos una diferència de velocitat abismal amb respecte als productes de la competència.
- **Baix cost.** MySQL està disponible de manera gratuïta, sota una llicència de codi obert, o per un preu reduït en forma de llicència comercial si així fos necessari per a la seva aplicació.
- **Facilitat** de configuració i aprenentatge.
- **Portabilitat.** Es pot utilitzar en una gran quantitat de sistemes Unix diferents així com amb Windows.
- **Accés al codi font.** Es pot obtenir i modificar el codi font de MySQL.

El llenguatge SQL és l'estàndard per accedir a bases de dades relacionals. Permet gestionar, controlar i definir bases de dades. Estem parlant d'un llenguatge normalitzat que ens permet treballar amb qualsevol tipus de llenguatge (Java, PHP, ASP) en combinació amb qualsevol gestor de bases de dades (MySQL, Microsoft SQL Server, etc).

L'SQL està format per comandes, clàusules, operadors i funcions d'agregat. Aquests elements es combinen en les instruccions per crear, actualitzar i manipular les bases de dades.



L'SQL proporciona una funcionalitat important més enllà de la simple consulta (o recuperació) de dades: assumeix el paper de llenguatge de definició de dades, llenguatge de definició de vistes i llenguatge de manipulació de dades. A més, en les últimes versions, permet la concessió i denegació de permisos, la implementació de restriccions d'integritat i controls de transacció, i l'alteració d'esquemes.

Per a més informació sobre MySQL: <http://www.mysql.com/>.

### **2.2.3. Arrencada remota**

Per tal d'aconseguir de forma remota que els ordinadors arrenquin, s'han de fer dues feines diferenciades: primer, com que els ordinadors estan apagats, s'han d'encendre remotament. Això ho fem usant la tecnologia Wake-on-LAN. Una vegada estan encesos, llavors se'ls ha d'enviar per xarxa la seva configuració i el menú d'arrencada. Per enviar-los la configuració de xarxa s'usa el protocol DHCP (o en alguns casos BOOTP). Per enviar un mini-*kernel* i el menú d'arrencada als PCs, s'usa el protocol TFTP. I, aglutinant tot això, hi ha l'entorn PXE que són una sèrie d'eines que implementen aquests protocols anteriors, necessaris per a l'arrencada remota.

En aquest capítol explicarem què són aquestes tecnologies, eines i protocols, i farem un resum del seu funcionament.

#### **2.2.3.1. Wake-on-LAN**

El Wake-on-LAN (WOL o bé WoL) és una tecnologia *hardware* i *software* estàndard d'Ethernet que permet arrencar remotament ordinadors que estan apagats enviant-los un paquet de xarxa especial; sempre i quan, aquests ordinadors estiguin preparats i activats per ser engegats per xarxa.

A l'abril de 1997, l'aliança Advanced Manageability Alliance entre Intel i IBM llançava la primera versió de la tecnologia Wake-on-LAN. Aviat, altres iniciatives industrials, com la Wired for Management (basada en Intel) donarien suport a aquest estàndard.

## 2. Descripció de l'entorn del sistema

---

Ara explicarem alguns detalls tècnics de com funciona la Wake-on-LAN. Aquests detalls, però, seran específics per a ordinadors del tipus compatible amb IBM PC. Com que als laboratoris hi ha aquest tipus d'ordinadors, llavors l'explicació ja ens serà útil.

El suport per a WoL, s'implementa a la placa base dels ordinadors. La placa base ha de tenir el connector WAKEUP-LINK connectat a la targeta de xarxa usant un cable especial de 3-vies (vegeu figura 2.3). Aquest cable fa que la targeta de xarxa estigui alimentada encara que el PC estigui apagat. Ara bé, si la placa base del PC suporta [PCI 2.2](#)<sup>2</sup> i la targeta de xarxa connectada a l'ordinador és compatible amb [PCI 2.2](#), llavors aquest cable WoL no cal perquè l'alimentació necessària per mantenir la targeta de xarxa en *stand-by* ja la proporciona el bus PCI. A més, la majoria de plaques base modernes amb la targeta de xarxa incorporada, també inclouen el suport per a WoL.

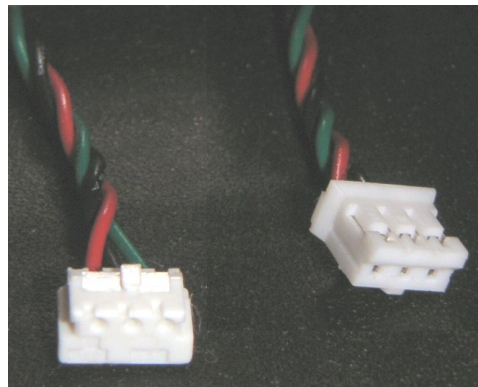


Figura 2.3: Cable i connectors de 3 pins especials per al Wake-on-LAN

Però perquè els ordinadors es puguin engegar remotament, també cal activar l'opció de Wake-on-LAN a la secció *Power Management* de la BIOS de la placa base perquè aquesta engegui el PC en cas de que la targeta de xarxa detecti el paquet especial de xarxa de Wake-on-LAN. I per mantenir la targeta de xarxa del PC alimentada quan el aquest estigui apagat, també pot ser necessari configurar alguna altra opció de l'ordinador.

El procés general de despertar un PC remotament es podria explicar així: el PC que volem engegar remotament està apagat, però amb la targeta de xarxa alimentada. La targeta de xarxa està escoltant la xarxa i esperant un paquet especial anomenat “*Magic Packet*” (Paquet Màgic). Llavors, algú envia aquest *Magic Packet* a la xarxa, ja sigui a tota la LAN o bé només a la sub-xarxa marcada per l'adreça de *broadcast*. En aquest moment, el PC que està escoltant, rep aquest paquet, comprova que sigui correcte i que vagi adreçat a ell. Si el *Magic Packet* és vàlid i ell n'és el destinatari, llavors s'engega.

El *Magic Packet* és un paquet de xarxa de tipus *broadcast* amb dades encapsulades que es transmet pels ports 0 (històricament el més usat), 7 o bé 9 (que està esdevenint el més usat en l'actualitat). Es pot enviar sobre gran quantitat de protocols de xarxa sense connexió (per exemple UDP o IPX), però el més usat és l'UDP. Les dades encapsulades que conté aquest paquet són, primerament una constant definida en hexadecimal com a FF FF FF FF FF FF, seguida de setze repeticions de l'adreça MAC del PC que volem engegar, i seguits, opcionalment, per una contrasenya formada per 4 o 6 bytes.

---

2 Per a més informació sobre PCI 2.2, podeu consultar la següent URL:  
[http://www.pcisig.com/specifications/conventional/conventional\\_pci/](http://www.pcisig.com/specifications/conventional/conventional_pci/)

### **2.2.3.2. BOOTP (*Bootstrap Protocol*)**

El BOOTP (*Bootstrap Protocol* en anglès) és un protocol de xarxa UDP usat pels clients de xarxa per obtenir la seva adreça IP automàticament. Normalment es fa servir durant el procés d'arrencada dels ordinadors o del sistema operatiu d'aquests.

Aquest protocol permet als ordinadors sense disc obtenir una adreça IP abans de carregar un sistema operatiu avançat. Històricament s'ha usat en estacions de treball UNIX (les quals també obtenien la localització de la seva imatge d'arrencada a través d'aquest protocol) i també per empreses per introduir una instal·lació pre-configurada de Windows a PCs acabats de comprar (típicament en un entorn de xarxa de Windows NT).

Originalment es necessitava fer servir un disquet d'arrencada per establir les configuracions de xarxa inicials, però després aquest protocol es va incorporar a la BIOS d'algunes targetes de xarxa (com la 3Com 3c905c) i en moltes plaques base modernes per permetre l'arrencada directa des de la xarxa.

Recentment, aquells amb interès per a PCs *media center* sense disc han mostrat un nou interès en aquest mètode per arrencar sistemes operatius Windows.

El DHCP és un protocol basat en BOOTP, més avançat, però més difícil d'implementar. Molts servidors DHCP també ofereixen suport per BOOTP.

### **2.2.3.3. DHCP (*Dynamic Host Configuration Protocol*)**

El DHCP (*Dynamic Host Configuration Protocol* en anglès) és un protocol de xarxa publicat a l'octubre de 1993 que permet als nodes d'una xarxa IP obtenir els seus paràmetres de configuració (com per exemple IP, màscara de xarxa, porta d'enllaç, adreça *broadcast* o bé adreça del servidor DNS) automàticament. Es tracta d'un protocol del tipus client-servidor en el qual generalment un servidor posseeix una llista d'adreces IP dinàmiques i les va prestant als clients a mesura que aquests les van demanant. Quan el període de préstec caduca, aquestes IPs van quedant lliures de nou. El servidor DHCP controla en tot moment qui ha estat o està en possessió de les IPs que ell proporciona i quant de temps les han tingudes, o quant de temps més les poden tenir.

## 2. Descripció de l'entorn del sistema

---

Sense DHCP, cada adreça IP s'ha de configurar manualment a cada ordinador i, si l'ordinador es mou a un altre lloc en una altra part de la xarxa, s'ha de configurar una altra adreça IP diferent. El DHCP permet a l'administrador supervisar i distribuir de forma centralitzada les adreces IP necessàries i assignar i enviar automàticament una nova IP si l'ordinador és connectat en un lloc diferent de la xarxa.

El protocol DHCP inclou tres mètodes d'assignació d'adreces IP:

- **Assignació manual o estàtica:** S'assigna una única adreça IP a cada ordinador diferent. L'administrador del sistema fa manualment una llista de parells IP-MAC. El DHCP, només donarà una adreça IP a l'ordinador amb una MAC que estigui a la seva llista. Se sol utilitzar quan es vol controlar l'assignació d'adreces IP per cada client, i evitar que es connectin clients no identificats.
- **Assignació automàtica:** S'assigna una adreça IP (escollida d'un rang d'IPs fixat per l'administrador del sistema) de forma permanent a cada ordinador client la primera vegada que fa la sol·licitud al servidor DHCP. Se sol utilitzar quan el nombre de clients no varia massa.
- **Assignació dinàmica:** És l'únic mètode que permet la reutilització dinàmica de les adreces IP. L'administrador de la xarxa determina un rang d'adreces IP que poden ser assignades. Cada vegada que un ordinador es connecta a la xarxa, sol·licita una adreça IP al servidor. Aquesta pot ser diferent cada vegada que un mateix ordinador es connecti a la xarxa. Usant aquest mètode d'assignació d'IPs es pot assignar un període de validesa a les IPs per a cada ordinador. Això facilita la instal·lació de nous ordinadors clients a la xarxa.

Algunes implementacions de DHCP poden actualitzar el DNS associat amb els servidors per reflectir les noves adreces IP mitjançant el protocol d'actualització de DNS.

El DHCP és una alternativa a altres protocols de gestió d'adreces IP de xarxa, com el BOOTP (Bootstrap Protocol). DHCP és un protocol més avançat, però avui dia encara s'usen ambdós normalment.

### 2.2.3.4. TFTP (*Trivial File Transfer Protocol*)

El TFTP (*Trivial File Transfer Protocol* en anglès, i que es traduiria com a Protocol Trivial de Transferència de Fitxers) és un protocol de transferència molt simple amb la funcionalitat d'una versió molt bàsica d'FTP. Va ser definit per primera vegada l'any 1980. El TFTP s'usa sovint quan un terminal X Window o qualsevol altre ordinador sense disc (client lleuger o *thin client* en anglès) arrenca des d'un servidor de xarxa.

Quan es començava a treballar sobre la *suite* de TCP/IP, sovint el TFTP era el primer protocol posat en marxa en els nous *hosts* perquè era molt simple.

Recentment, el TFTP s'ha usat en cucs (gusanos, o *worms* en anglès) informàtics, com el Blaster, com a mètode de difusió i infecció de nous ordinadors.

A continuació expliquem breument alguns detalls del TFTP:

- Com a protocol de transport usa UDP (port 69), a diferencia de l'FTP que usa el port 21 TCP.
- No pot llistar el contingut des directoris.
- No té mecanismes d'autenticació o xifrat.
- S'usa per llegir o escriure arxius d'un servidor remot.
- Suporta tres modes diferents de transferència: *netascii*, *octet* i *mail*. Els dos primers corresponen als modes *ascii* i *imatge (binari)* del protocol FTP respectivament. El tercer està obsolet i gairebé ja no s'usa.
- A causa de la carència de la seguretat, és perillós usar-lo sobre internet obert. Així doncs, generalment el TFTP només s'usa en xarxes locals o privades.

Com que el TFTP usa el protocol de transport UDP, no hi ha una definició formal de sessió de client-servidor. És el mateix TFTP que ha de proveir el seu propi suport de transport i de sessió. Malgrat tot, cada arxiu transferit via TFTP constitueix un intercanvi independent de paquets i existeix una relació informal de client-servidor entre l'ordinador que inicia la comunicació i el que contesta. Ara explicarem, doncs, com funciona una sessió de TFTP:

**Pas 1.** La màquina A, que inicia la comunicació, si vol escriure, envia un paquet WRQ (*Write Request* en anglès o petició d'escriptura) a la màquina S; i si vol rebre, envia un paquet RRQ (*Read Request* en anglès, o petició de lectura) a la màquina S. En ambdós casos aquest paquet conté el nom de l'arxiu i el mode de transferència.

**Pas 2.** La màquina S, si ha rebut un WRQ, contesta amb un paquet ACK (*Acknowledgement* en anglès o confirmació); i si ha rebut un RRQ contesta directament amb un paquet de dades. Aquests paquets també serveixen per informar a la màquina A del port (de la S) al que haurà d'enviar els paquets que queden en cas de WRQ; els paquets ACK en cas de RRQ.

**Pas 3.** La màquina d'origen (la que envia) envia paquets de dades numerats a la màquina de destí (la que rep). Tots aquests contenen 512 bytes de dades, excepte l'últim. La màquina de destí contesta a cada paquet de dades amb paquets ACK numerats.

**Pas 4.** El paquet de dades final ha de contenir menys de 512 bytes de dades para indicar que és l'últim. Si la mida de l'arxiu transferit és un múltiple exacte de 512 bytes, la màquina d'origen envia un paquet final que conté 0 bytes de dades.

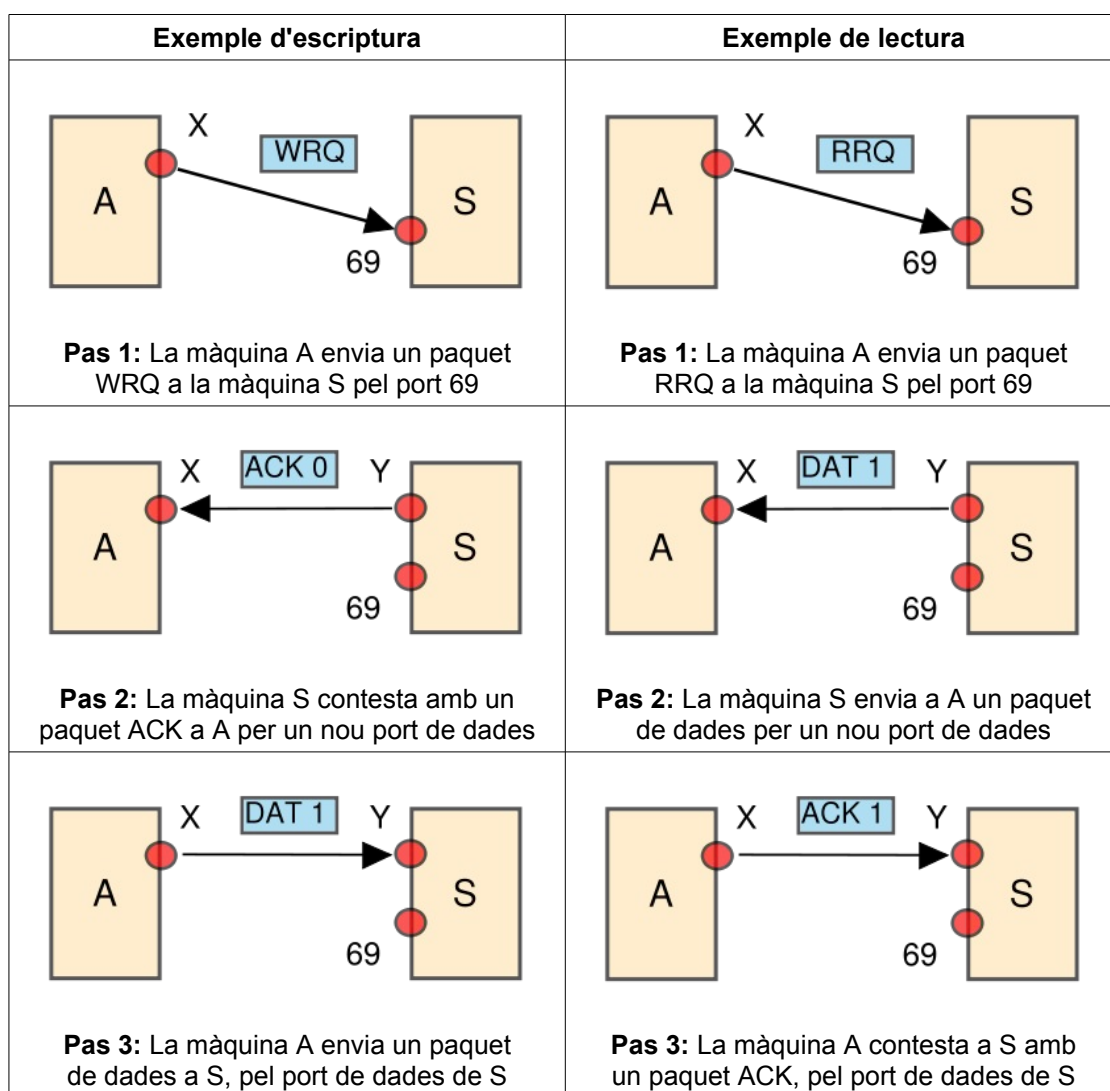


Figura 2.4: Esquemes de funcionament del protocol TFTP, d'escriptura i de lectura

### 2.2.3.5. PXE

El PXE (*Preboot Execution Environment* en anglès, que es traduiria com a Entorn d'Execució Previ a l'Arrencada) és un entorn que inclou i implementa totes les eines necessàries perquè un ordinador client pugui arrencar usant la seva targeta de xarxa. En arrencar, l'ordinador client es descarrega la configuració de xarxa i els arxius necessaris per arrencar, independentment dels sistemes operatius o dels dispositius d'emmagatzematge de dades (com, per exemple, discs durs) que tingui instal·lats.

O sigui, que un servidor PXE, implementa i usa un protocol que és una barreja de DHCP i de TFTP: és capaç d'enviar la configuració de xarxa als clients que ho sol·licitin (de fet, el protocol que usa es diu *Proxy DHCP* i és com el DHCP però amb alguns paràmetres afegits), i també és capaç d'enviar als clients un *mini-kernel* d'arrencada (que es diu NBP, *Network Bootstrap Program* en anglès) usant el protocol TFTP.

El PXE va ser introduït per Intel a finals dels 90 com una part del Wired for Management (WfM). Aquest era un *hardware* que permetia introduir programes al disc dur d'un ordinador nou, a partir d'un altre ordinador principal. L'aparell WfM també es podia fer servir per actualitzar *software* i monitoritzar l'estat dels sistemes remotament.

Entenem per client PXE l'ordinador que té una targeta de xarxa compatible amb PXE (targeta de xarxa amb un *firmware* amb funcions especials de PXE) i necessita un servidor PXE per arrencar.

Quan un client PXE s'engega, primer de tot, necessita una adreça IP i informació sobre els servidors d'arrencada PXE disponibles. Per obtenir aquesta i altra informació, intenta localitzar un servei de redirecció PXE (*Proxy DHCP*) enviant un paquet *DHCPDISCOVER*, ampliat amb paràmetres específics de PXE (*extended DHCPDISCOVER*) al port d'un servidor DHCP (port 67/UDP). Quan un servidor PXE rep un *DHCPDISCOVER*, consulta si aquest paquet té els paràmetres que identifiquen que prové d'un client PXE. Si el *DHCPDISCOVER* no prové d'un client PXE, el servidor PXE l'ignora. En canvi, si prové d'un client PXE, el servidor li envia la configuració de xarxa demanada al port de client DHCP (port 68/UDP).

Una vegada el client té una IP i sap quins servidors d'arrencada hi ha, el demana a un d'aquests servidors d'arrencada el nom d'un fitxer que conté el programa que usará el client PXE per arrencar. Aquest programa es diu NBP (*Network Bootstrap Program* en anglès). Després, aquest client usa el protocol TFTP per descarregar-se aquest fitxer, desar-lo a la seva RAM, i executar-lo. Si només hi ha un sol NBP per a tots els clients PXE, aquests se'l poden descarregar usant el protocol BOOTP, sense necessitar cap *Proxy DHCP* ni cap servidor d'arrencada PXE.

El Protocol de client-servidor de PXE es va dissenyar perquè complís els següents requeriments:

- El PXE no ha d'interferir en una xarxa amb un entorn de DHCP existent.
- El PXE s'ha de poder integrar completament a serveis de DHCP estàndards.
- Tots els serveis (DHCP, *Proxy DHCP* o servidor d'arrencada) s'han de poder implementar independent o en qualsevol combinació d'ells.

Com que servidor PXE ha de poder conviure amb un servidor DHCP normal i aquests 2 no poden compartir el port 67/UDP, si ja hi ha un servei de DHCP quan s'instal·la el servei PXE, aquest últim espera el *DHCPDISCOVER* pel port 4011/UDP. Llavors, el servidor DHCP estàndard, ha d'enviar als clients una combinació especial d'opcions PXE al seu *DHCPOFFER* perquè els clients sàpiguin que al mateix host hi ha un servidor PXE.

### 2.2.4. Tecnologia d'imatges i servidor d'imatges

En aquest apartat, explicarem què és la tecnologia d'imatges, què és un servidor d'imatges i en veurem un en concret que és el Rembo.

#### 2.2.4.1. Introducció al concepte d'imatges de software

La gent que porta laboratoris o aules informàtiques, ja fa temps que usa la tecnologia d'imatges per mantenir el *software* dels ordinadors d'aquests de manera fàcil i ràpida.

Al principi, quan no existia la tecnologia de les imatges, el *software* de cada ordinador d'un laboratori s'havia d'instal·lar per separat. Per cada ordinador, s'havia d'instal·lar el sistema operatiu, configurar-lo i instal·lar-hi tots els programes un per un. Tenint en compte que moltes aules informàtiques tenien 10 ordinadors o més, això suposava un gran volum de feina. A més, per fer actualitzacions de *software* també s'havia d'anar ordinador per ordinador. I a sobre, si el programari d'un ordinador es malmetia, s'havia d'instal·lar de nou des de zero. Lògicament això comportava una dedicació de temps i recursos humans molt gran.

Part d'aquesta feina es va eliminar instal·lant aplicacions en un servidor, i així tots els ordinadors dels laboratoris podien usar, a través de la xarxa, el programa instal·lat al servidor. Això també es feia perquè els ordinadors dels laboratoris tenien poca capacitat de càlcul, i en canvi els servidors eren màquines força més potents. Però això comportava carregar més l'ordinador que servia aplicacions. Imaginem-nos una universitat amb més de 100 ordinadors als seus laboratoris! De vegades, els servidors d'aplicacions anaven sobresaturats.

Amb el temps, quan els ordinadors personals van augmentar considerablement la seva potència de càlcul i sortia més a compte comprar-ne diversos que no pas un gran servidor. Llavors els servidors d'aplicacions van anar una mica de baixa (com a mínim en l'entorn que ja conec) i els ordinadors dels laboratoris van tornar a tenir molt de *software* instal·lat, cosa que resultava molt costós de mantenir.

A partir d'aquest instant, van aparèixer els programes que feien imatges del *software* dels ordinadors. Podríem definir la paraula 'imatge' (de *software*) com a una còpia exacta (desada en un fitxer o més depenent de la mida o del programa d'imatges) de tot contingut d'alguna partició o d'algun disc dur d'un ordinador.

Si tenim un laboratori amb molts ordinadors idèntics (o gairebé), podem instal·lar el *software* en un d'ells i configurar-lo perquè funcioni correctament. Una vegada instal·lat, usant un programa d'imatges, podem crear-ne una imatge, desar-nos-la i restaurar-la als altres PCs del



laboratori. Restaurar una imatge vol dir copiar tot el contingut d'una imatge al disc dur o partició d'un ordinador, de tal manera que aquest ordinador tingui instal·lat exactament el mateix *software* que l'ordinador del qual procedeix la imatge.

El procés de creació i de restauració d'imatges pot ser bastant senzill i ràpid si es coneix el programa d'imatges que s'usa.

Amb aquesta tecnologia és possible instal·lar un sol ordinador d'una aula informàtica, crear-ne la imatge, i restaurar-la a tots els altres PCs del laboratori.

A partir d'aquí, ens apareix el problema d'on emmagatzemar les imatges i de com restaurar-les a tots els PCs d'un laboratori. El que es feia, i segurament encara es fa en alguns llocs, era tenir una col·lecció de CDs o DVDs autoarrencables, que contenien el programa d'imatges i alguna imatge. Llavors, només amb aquest CD o DVD ja es podia restaurar la imatge dels ordinadors d'un laboratori. Això sí, s'havia d'anar un per un, però el procés de restauració podia anar-se executant sol, una vegada havia començat.

Però després va aparèixer la idea de servidor d'imatges que veurem tot seguit.

### **2.2.4.2. Servidor d'imatges**

Podem definir com a servidor d'imatges, un programa d'imatges instal·lat en un ordinador que conté una sèrie d'imatges d'un o més laboratoris al seu disc dur, i que és capaç de servir aquestes imatges a través d'una xarxa informàtica.

Classifiquem els servidors d'imatges en dos tipus:

- Els que necessiten un programa a l'ordinador client que es posi en contacte amb els servidor i negociï la transferència de la imatge. En aquest cas, algú ha d'anar als PCs dels laboratoris, arrencar-los amb aquest client d'imatges, i configurar-ne la xarxa.
- Els que són capaços d'enviar aquest programa client d'imatges a través de la xarxa usant l'entorn PXE als ordinadors dels laboratoris, tan bon punt aquests s'engeguin. Aquest tipus de servidors són molt útils perquè, com que poden enviar en programa que s'executa automàticament als ordinadors dels laboratoris quan arrenquen, doncs se'ls pot enviar un menú que apareix a tots els ordinadors dels laboratoris i que permet escollir si es vol arrencar-lo o bé restaurar-ne la imatge. Això ho poden fer els mateixos usuari dels PCs dels laboratoris encara que no tinguin grans coneixements d'informàtica ni d'imatges ni de xarxes.

### 2.2.4.3. Rembo

El **Rembo** és un programa servidor d'imatges de l'últim tipus dels que hem vist: és capaç de servir imatges a través de la xarxa, però també és capaç d'enviar al client d'imatges un petit programa que aquest executarà en arrencar. Aquest programa d'arrencada, té la possibilitat de presentar un menú inicial que permeti als usuaris NO experts en informàtica arrencar un sistema operatiu, restaurar alguna imatge o fer altres coses com per exemple formatejar el disc dur dels ordinadors dels laboratoris.

L'estiu del 1996, una gent del departament *The Computer Science Department* de la *CUI, University of Geneva* va implementar un programa lliure que era un servidor d'imatges i es deia **BpBatch**. Aquest programa va agafar molta fama i va esdevenir força popular entre els administradors de *software* de laboratoris informàtics.

L'any 1999 el mateix equip de gent que havia desenvolupat el **BpBatch**, va crear una l'empresa Rembo Technology SaRL. Aquesta empresa va treure al mercat la versió comercial de **BpBatch** que es deia **Rembo**. Del **Rembo**, se'n va fer molt popular la versió 2.0, que encara s'usa avui en dia, i la versió 4.0 que és la que tenim al CBL.

L'any 2006, l'empresa IBM va comprar **Rembo** i el va incloure en el seu producte IBM Tivoli®. La part de Tivoli que correspondria a l'antic **Rembo** 4.0 es diu IBM Tivoli® Provisioning Manager for OS Deployment<sup>3</sup>.

Les antigues webs del **BpBatch** i de **Rembo** eren [www.bpbatch.org](http://www.bpbatch.org) i [www.rembo.com](http://www.rembo.com), però ara, aquestes dues et redirigeixen cap a la pàgina d'IBM.

---

<sup>3</sup> Per a més informació sobre aquest producte podeu consultar l'URL <http://www-306.ibm.com/software/tivoli/products/prov-mgr-os-deploy/>

### 2.3. Treballs relacionats

Existeixen diversos programes per restaurar imatges com, per exemple, el DriveImage (<http://www.drive-image.com/>) que és *shareware*, però que no té la funcionalitat de servidor d'imatges. Només serveix per fer la creació/restauració d'imatges localment.

També hi ha el Symantec Ghost (<http://www.symantec.com/ghost/>) que és un programa comercial molt potent que sí que té la part de servidor d'imatges (es diu GhostCast Server) i ja fa anys que amb un petit client posat a cada ordinador permet restauracions d'imatges d'ordinadors de laboratoris. També inclou la capacitat de restaurar tots els PCs de l'aula alhora usant la tècnica *multicast*. Sembla ser que les últimes versions del Symantec Ghost també incorporen un mètode per enviar el fitxer d'arrencada automàticament sense haver d'anar als ordinadors clients un per un.

Pel que fa al Rembo, nosaltres, en aquest projecte, només n'usem una petita part. Realment és una eina molt completa i molt potent que permet enviar tota la configuració i programes necessaris perquè els ordinadors arrenquin directament des de la xarxa, usant el PXE. Però també permet engegar els ordinadors apagats de manera remota. A més, també s'encarrega d'emmagatzemar de manera molt eficient totes les imatges i altres dades necessàries per funcionar.

I el que fa el Rembo una eina encara més potent, és que té una interfície web amb validació inclosa per poder-lo gestionar des de qualsevol lloc.

Ara bé, el Rembo té els inconvenients que és un *software* comercial i que, per gestionar-lo, s'han de tenir una sèrie de coneixements una mica avançats del seu funcionament.

El que fa el nostre sistema bastant acceptable, és que implementa totes aquestes capacitats usant *software* lliure (exceptuant-ne la part que usem del Rembo) i que ofereix una interfície web molt amigable i que es pot usar sense tenir cap coneixement d'alt nivell de les eines que s'usen. A més, la base de dades pot ser útil per si sola inclòs si no s'usa la part de restauració d'imatges. Per acabar, també és molt útil que la aplicació web mostri l'estat dels PCs dels laboratoris a qualsevol usuari, encara que no estigui validat, perquè així pot preveure millor les classes o les pràctiques.



### 3. Anàlisi de requisits

3.1. Introducció.....	43
3.2. Requisits funcionals.....	44
3.2.1. Validació.....	44
3.2.1.1. Usuaris no validats.....	44
3.2.1.2. Usuaris validats.....	45
3.2.2. Consulta a l'inventari (de PCs, laboratoris i edificis).....	45
3.2.3. Gestió d'estats (de PCs i laboratoris).....	46
3.2.4. Gestió d'imatges.....	46
3.2.5. Manteniment (de PCs i laboratoris).....	47
3.3. Requisits no funcionals.....	47
3.3.1. Interfície.....	48
3.3.2. Qualitat del software.....	48
3.3.3. Càrrega.....	49
3.3.4. Tecnològics.....	50
3.3.4.1. Comprovació del hardware disponible.....	50

#### 3.1. Introducció

En el primer capítol d'aquest treball, hem descrit els objectius que ha de cobrir aquest sistema. Ara, a la fase d'anàlisi de requisits farem un estudi tècnic de les necessitats i objectius del projecte. Concretarem els requeriments del sistema fent servir les eines d'anàlisi UML.

Tenim dos tipus de requisits o requeriments:

- **Requisits funcionals:** Són els que expliquen el comportament que ha de tenir el nostre sistema: cada requisit funcional expressa una relació entre entrades i sortides del sistema. És a dir, els requisits funcionals especifiquen les sortides que ha de produir el nostre sistema a partir d'unes entrades determinades i les operacions necessàries per aconseguir-les. A més, també especifiquen com s'ha de comportar el sistema en situacions anormals.

### 3. Anàlisi de requisits

---

- **Requisits no funcionals:** Aquests requisits no són conseqüència directa de cap necessitat funcional dels usuaris, sinó que, normalment, són restriccions imposades pel client o bé pel propi problema, però que afecten al disseny d'aquest sistema.

#### 3.2. *Requisits funcionals*

*A priori*, distingim al nostre sistema les següents àrees d'actuació que permeten classificar els requisits funcionals de manera que resulti més clar identificar-los i entendre'ls:

- Validació
- Consulta a l'inventari (de PCs, laboratoris i edificis)
- Gestió d'estats (de PCs i laboratoris)
- Gestió d'imatges
- Manteniment (de PCs i laboratoris)

##### 3.2.1. Validació

El sistema ha de permetre que un usuari es validi impedir que qualsevol persona NO autoritzada usi algunes funcions del projecte que només haurien de poder utilitzar les persones autoritzades. Aquesta validació provoca que els usuaris del projecte es divideixin en dos tipus:

### **3.2.1.1. Usuaris no validats**

Els usuaris d'aquest tipus només han de poder accedir a les prestacions de visualització, o sigui, només han de poder efectuar consultes a l'inventari de PCs, laboratoris i edificis. A través d'aquest inventari, han de poder veure el llistat d'edificis i de laboratoris, i la distribució i les dades dels PCs del CBL. A més, els usuaris no validats han de poder veure l'estat i l'historial d'estats dels laboratoris i dels PCs del CBL.

Perquè un usuari formi part d'aquest grup, no ha de fer res: tots els usuaris que no s'han validat formen part d'aquest grup d'usuaris i, com que no s'han validat, no tenen ni nom d'usuari ni contrasenya.

### **3.2.1.2. Usuaris validats**

Els usuaris d'aquest tipus han de poder accedir a totes les prestacions del projecte que no hi puguin accedir els usuaris no validats, o sigui, han de poder gestionar (modificar) l'estat dels PCs i dels laboratoris, gestionar (crear, modificar i suprimir) les imatges dels PCs del CBL, o bé executar el manteniment dels PCs o dels laboratoris.

Perquè un usuari formi part d'aquest grup, ha d'estar autoritzat pels Serveis Tècnics del CBL (d'ara endavant ST-CBL), els quals, en el moment de l'autorització, li donaran un nom d'usuari i una contrasenya que haurà d'introduir quan el sistema d'autenticació del Projecte li demani.

#### **3.2.2. Consulta a l'inventari (de PCs, laboratoris i edificis)**

Aquest inventari ha de contenir les dades dels edificis, dels laboratoris i dels PCs emmagatzemades a la base de dades del primer ítem. Aquestes dades només les podrem consultar i s'hauran de mostrar, de manera senzilla i amigable.

Si estem consultant edificis o bé laboratoris, aquests s'hauran de mostrar en forma de llistat. En canvi, si estem consultant els PCs d'un laboratori, s'haurà de mostrar la distribució d'aquests dintre del laboratori. Per acabar, si estem consultant les dades d'un sol PC, aquestes s'hauran de mostrar en format fitxa.

#### **3.2.3. Gestió d'estats (de PCs i laboratoris)**

El sistema ha de permetre gestionar, de manera senzilla i amigable, els estats dels PCs i dels laboratoris del CBL. És a dir, el sistema ha de permetre mostrar i modificar l'estat cada un dels PCs i dels laboratoris. A més, s'ha de poder modificar l'estat dels PCs o dels laboratoris, seleccionant-ne només un, o bé seleccionant-ne diversos i modificar amb una sola operació l'estat de tots ells.

Quan gestionem els estats dels PCs, la paraula estat s'ha de referir a si estan operatius, fora de servei, o bé en manteniment, per exemple. En canvi, quan gestionem els estats dels laboratoris, la paraula estat s'ha de referir només a si estan oberts o bé tancats.

A més, hi ha d'haver un històric de l'estat de cada PC i de cada laboratori del CBL i el sistema ha de permetre mostrar aquest historial seleccionant-ne només un, o bé seleccionant-ne diversos i mostrar amb una sola operació l'historial dels estats de tots ells.



### **3.2.4. Gestió d'imatges**

El sistema ha de permetre gestionar, de manera senzilla i amigable, les imatges dels PCs del CBL. És a dir, el sistema ha de permetre crear, modificar i esborrar imatges de PCs. A més, també ha de permetre assignar aquestes imatges a un o més PCs o laboratoris, tenint en compte que el sistema ha de ser capaç d'imitar la distribució actual de les imatges pels laboratoris del CBL. Aquesta distribució actual és la següent:

- Hi ha laboratoris en què tots els seus PCs tenen la mateixa imatge (o les mateixes imatges si n'hi ha més d'una per PC), i cap d'elles no és en cap altre PC de cap altre laboratori.
- Hi ha laboratoris en què tots els seus PCs tenen la mateixa imatge (o les mateixes imatges si n'hi ha més d'una per PC), i, com a mínim una, també és a altres PCs d'altres laboratoris.
- Hi ha laboratoris en què els seus PCs tenen imatges diferents, i cap d'aquestes no és en cap altre PC de cap altre laboratori.
- Hi ha laboratoris en què els seus PCs tenen imatges diferents, i com a mínim una també és a altres PCs d'altres laboratoris.

### **3.2.5. Manteniment (de PCs i laboratoris)**

El sistema ha de permetre simular, de manera senzilla i amigable, la restauració de les imatges d'un o més PCs o laboratoris. És a dir, si només volem simular la restauració d'un sol PC, s'ha de poder seleccionar quina o quines imatges d'aquest PC volem restaurar; i després permetre, amb una sola comanda, començar la simulació de la restauració d'aquestes. Si volem simular la restauració de més d'un PC, s'ha de poder seleccionar quin o quins PCs volem restaurar; i després permetre, amb una sola comanda, començar la simulació de la restauració de totes les imatges dels PCs seleccionats. I per acabar, si volem simular la restauració d'un o més laboratoris, s'ha de poder seleccionar quin o quins laboratoris volem restaurar; i després permetre, amb una sola comanda, començar la simulació de la restauració de totes les imatges de tots els PCs dels laboratoris seleccionats.

### 3. Anàlisi de requisits

---

Per poder dur a terme aquestes restauracions, cal que el projecte tingui la capacitat de controlar remotament l'arrencada dels PCs dels diversos laboratoris del CBL. L'arrencada d'aquests PCs s'ha de fer de manera automàtica cada vegada que es doni l'ordre de restaurar un PC. Una vegada que els PCs s'engeguin de manera remota, automàticament ha de començar-ne la restauració.

A més, les restauracions s'han de poder fer de manera interactiva (l'usuari validat de l'aplicació és el que les inicia) o bé que l'usuari validat de l'aplicació pugui programar-ne el moment d'inici amb anterioritat i pugui fixar-ne repeticions periòdiques i automàtiques.

Finalment, quan aquestes restauracions s'acabin, el sistema ha de reportar l'estat de les restauracions i emmagatzemar-lo (el report) a la base de dades on s'emmagatzemin les dades i els estats dels PCs del CBL.

#### **3.3. Requisits no funcionals**

Com hem introduït abans, aquests requisits fan referència a qüestions alienes a les funcionalitats o processos del projecte. Defineixen les qualitats generals que ha de tenir el sistema per funcionar correctament. Tot i que són implícits al projecte, poden resultar fonamentals pel bon funcionament d'aquest.

És sabut que hi ha conflictes entre molts dels requisits no funcionals desitjables per a tot sistema, i també entre els exposats a continuació. De totes maneres, a l'hora de dissenyar i implementar el projecte, tindrem presents els més convenients segons les característiques del nostre sistema i establim prioritats entre ells.

### 3.3.1. Interfície

La interfície amb l'usuari proposada tindrà les següents característiques:

- **Senzillesa, claredat i amigabilitat.** La interfície haurà de ser suficientment senzilla i intuïtiva perquè la puguin fer servir persones sense coneixements avançats d'informàtica. Les pàgines seran el menys denses possible i ben estructurades. A més, els menús i el text hauran de ser clars; i els títols hauran de destacar. En general, la aplicació web haurà de ser amigable i còmoda.  
  
Cal remarcar que aquest requisit és molt important pel nostre sistema perquè desitgem facilitar al màxim la interacció de l'usuari amb el sistema.
- **Transparència per a l'usuari.** Un usuari normal de l'aplicació no necessitarà conèixer l'estructura de la base de dades (taules, relacions, ...) que dona suport a l'aplicació, ni cap llenguatge de programació per usar l'aplicació web.

### 3.3.2. Qualitat del *software*

Factors de qualitat que haurà de complir el *software*:

- **Eficiència.** L'aplicació utilitzarà els mínims recursos possibles, per tal de no sobrecarregar el sistema.
- **Flexibilitat.** Haurà de ser relativament senzill modificar parts del programari. Com que és un prototip, aquest aspecte és rellevant, perquè ha d'estar preparat per si es modifica.
- **Integritat.** El sistema haurà d'evitar la introducció de redundàncies o inconsistències a la base de dades.
- **Seguretat 1: identificació i reconeixement dels usuaris.** Les parts no públiques de l'aplicació web (o sigui, les que requereixin autenticació) només seran accessibles per la gent autoritzada pels Serveis Tècnics del CBL (seguretat externa). Caldrà poder definir a quines funcionalitats de l'aplicació té accés cadascun dels usuaris de la

### 3. Anàlisi de requisits

---

mateixa, i a quines no; i assegurar que cap usuari pugui executar cap funció per a la que no hi té accés (seguretat interna). O sigui, caldrà tenir en compte els següents punts:

- ◊ **Seguretat externa:** Comprovar que la gent que vulgui accedir a les parts del sistema que requireixin autenticació estigui autoritzat.
- ◊ **Seguretat interna:** Definir els permisos que té cada usuari validat del sistema.
- **Seguretat 2: confidencialitat de les dades.** Les dades que s'emmagatzemaran a la base de dades seran confidencials. Per tant s'ha d'assegurar que cap persona no autoritzada hi pugui accedir. Això ja ho garantirà el propi SGBD.  
  
A més, com que el *software* és una aplicació web, i com a tal basada en internet, hem de garantir que les dades que viatgin per la xarxa puguin ser consultades i/o retocades per qualsevol persona sense autorització. Per tant, usarem el protocol segur HTTPS sobre SSL (*Secure Socket Layer*) perquè les dades viatgin encriptades per la xarxa.  
  
Per acabar, les paraules de pas s'hauran d'emmagatzemar a la base de dades de manera encriptada.
- **Portabilitat.** Hauria de ser relativament fàcil canviar el *software* de plataforma. Per exemple, canviar-lo de sistema operatiu. El llenguatge de programació escollit (PHP) no presenta massa problemes per canviar de plataforma.
- **Eficàcia i fiabilitat.** Desitgem obtenir els resultats esperats de totes les funcionalitats del sistema amb el mínim d'errors possible. En qualsevol cas, la resposta del sistema sempre ha de ser coherent i adient.
- **Reusabilitat.** Part del sistema s'hauria de poder usar en altres sistemes si és necessari.

#### 3.3.3. Càrrega

És obligatori que el sistema presenti uns bons resultats de càrrega. És un punt clau per a l'èxit del sistema.

Tenim tres factors importants que suposen restriccions:

- **Requisits sobre les característiques d'execució.** En aquest punt hem de tenir en compte que l'aplicació podrà tenir bastants usuaris simultanis connectats al sistema.

- **Requisits sobre el comportament de l'execució del sistema.** Hem de garantir que el temps de resposta pels clients connectats sigui acceptable. Així, el temps que tardarà el sistema en realitzar les diferents funcionalitats serà el mínim possible, evitant esperes llargues als usuaris. Suposem que el temps d'espera màxim per a qualsevol pàgina de l'aplicació és de 10 segons. Si se superés aquest temps d'espera en alguna de les pàgines, l'aplicació no compliria aquest requisit de rendiment, i no seria un servei usable. Hem escollit aquests 10 segons com a temps màxim d'espera gràcies a la informació que aporta l'article *Web Page Response time 101* d'Alberto Saviola [6].
- **Requisits sobre la taxa d'errors del servidor.** Generalment, no considerem admissible un percentatge d'errors superior a l'1% del total de transaccions executades durant les proves. Hem escollit aquest percentatge d'error gràcies a l'article *Una introducción a las pruebas de prestaciones* de José María Morales Vázquez [7].

### 3.3.4. Tecnològics

El sistema ha de funcionar a través de web basat en protocols estàndards d'Internet.

El CBL imposa en aquest aspecte dues condicions: per una part, s'haurà de desenvolupar el sistema perquè pugui funcionar amb el *hardware* i el sistema operatiu de què disposa actualment: un ordinador amb Linux que és el servidor webm, i un altre ordinador amb Linux que és el servidor d'imatges. De totes maneres, per executar aquest sistema prototipus, no usarem cap d'aquests dos ordinadors perquè estan en explotació, sinó que usarem un altre ordinador amb tots els servidors necessaris instal·lats en ell.

Per l'altra part, el *software* que hem de fer servir per dur a terme les restauracions de les imatges dels PCs ha de ser el Rembo v4.0 perquè totes les imatges del CBL existents estan fetes amb Rembo v4.0.

D'altra banda, les tecnologies utilitzades en el desenvolupament del sistema hauran de ser considerades estàndards i no podran dependre de cap plataforma o propietari. S'usarà UML (*Unified Modeling Language*) en l'especificació i disseny del *software*.

Qualsevol navegador web actual hauria de ser suficient pel funcionament correcte de l'aplicació, tenint en compte, però, que l'aplicació web està pensada per ser utilitzada en dispositius amb resolució de pantalla de 1024x768 píxels, tot i que ha de ser fàcilment adaptable a resolucions superiors.

Per arrencar els PCs de manera remota, el *hardware* d'aquests ha de tenir unes funcionalitats concretes. Per tant, hem de comprovar que els PCs dels laboratoris les tenen.

#### **3.3.4.1. Comprovació del hardware disponible**

Per comprovar el maquinari del tots els PCs del CBL, hem anat laboratori per laboratori i hem constatat que:

- Tots els laboratoris informàtics del CBL, estan proveïts d'ordinadors personals (PCs) amb microprocessadors de 32 bits i arquitectura i386.
- Tots els PCs del CBL tenen la capacitat d'arrencar per xarxa i la funcionalitat de Wake-on-LAN.
- Totes les targetes de xarxa dels PCs del CBL incorporen el protocol PXE i tenen la funcionalitat de Wake-on-LAN.

A més també hem pogut veure que, com que al CBL hi ha diversos tipus de *hardware*, això requereix crear, com a mínim, tantes imatges com configuracions de *hardware* existeixin.

## 4. Especificació i disseny

4. Especificació i disseny.....	53
4.1. Especificació.....	54
4.1.1. Casos d'ús.....	54
4.1.1.1. Actors.....	54
4.1.1.2. Diagrames dels casos d'ús.....	55
Casos d'ús de validació.....	55
Casos d'ús de consulta a l'inventari (de PCs, laboratoris i edificis).....	56
Casos d'ús de gestió d'estats (de PCs i laboratoris).....	56
Casos d'ús de gestió d'imatges.....	57
Casos d'ús de manteniment (de PCs i laboratoris).....	57
4.1.1.3. Especificació dels casos d'ús.....	58
Casos d'ús de validació.....	58
Casos d'ús de consulta a l'inventari (de PCs, laboratoris i edificis).....	59
Casos d'ús de gestió d'estats (de PCs i laboratoris).....	61
Casos d'ús de gestió d'imatges.....	64
Casos d'ús de manteniment (de PCs i laboratoris).....	66
4.2. Disseny.....	69
4.2.1. Diagrama de classes.....	70
4.2.2. Disseny lògic de la base de dades.....	70
4.2.3. Diagrames d'activitats.....	71
4.2.3.1. Diagrama d'activitats per a usuaris no validats.....	72
4.2.3.2. Diagrama d'activitats per a usuaris ja validats.....	72
4.2.4. Arquitectura del software.....	73
4.2.4.1. Patrons arquitectònics i de disseny.....	74
4.2.4.2. El patró Model-Vista-Controlador i la seva adaptació per a web.....	75
4.2.4.3. Controlador façana.....	76
4.2.5. Diagrama de classes amb el patró MVC.....	77
4.2.6. Diagrames de seqüència dels casos d'ús.....	79
4.2.6.1. Mostra edificis.....	79
4.2.6.2. Mostra laboratoris.....	79
4.2.6.3. Mostra PCs.....	80
4.2.6.4. Mostra PC.....	81
4.2.6.5. Canvia estat laboratoris.....	81
4.2.6.6. Canvia estat PCs.....	82
4.2.6.7. Canvia estat PC.....	82
4.2.6.8. Mostra historial estat laboratoris.....	83
4.2.6.9. Mostra historial estat PCs.....	83
4.2.6.10. Mostra historial estat PC.....	83
4.2.6.11. Crea imatge.....	84
4.2.6.12. Modifica imatge.....	84

## 4. Especificació i disseny

---

4.2.6.13. Esborra imatge.....	85
4.2.6.14. Restaura laboratoris.....	87
4.2.6.15. Restaura PCs.....	89
4.2.6.16. Restaura PC.....	90
4.2.7. Disseny de la interfície.....	91

Usant UML, la frontera entre especificació i disseny és, de vegades, difusa perquè que s'especifica i es dissenya a la vegada. Tot i així, hem considerat que el model dels casos d'ús és clarament especificació, perquè que quan els descrivim expliquem el que fan i no com ho fan. La resta de models que usem, ja els hem considerat disseny.

### 4.1. Especificació

#### 4.1.1. Casos d'ús

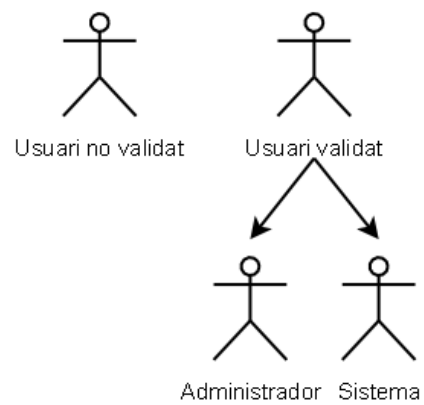
##### 4.1.1.1. Actors

En els requisits d'aquest projecte, ja hem vist dos tipus d'usuaris, que aquí esdevenen actors: són **Usuari no validat** i **Usuari validat**. Però en els casos d'ús, nosaltres no parlarem d'Usuaris validats, sinó que els subdividirem en **Administrador** i **Sistema**.

L'actor **Administrador** és l'usuari que s'ha validat manualment i té privilegis per accedir a les zones restringides d'aquest projecte. L'actor **Sistema** també l'hem considerat un **Usuari Validat** perquè és un actor que té privilegis per fer coses que els usuaris no validats no poden fer, però no s'ha validat manualment. Podríem dir que l'ha validat el programador del sistema.

Nosaltres només usarem tres tipus d'actors: **Usuari no validat**, **Administrador** i **Sistema**.

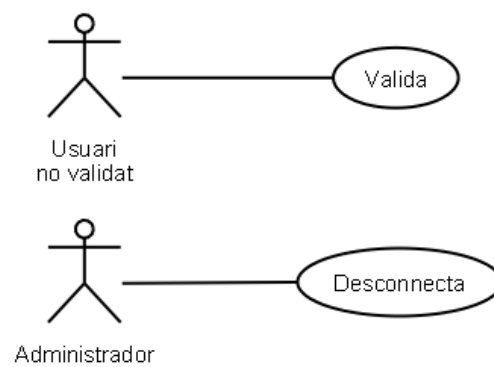




*Figura 4.1.1: Actors dels casos d'ús*

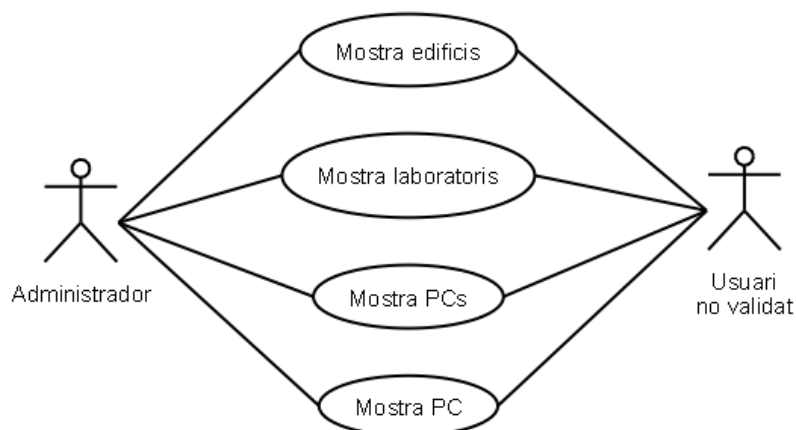
#### 4.1.1.2. Diagrames dels casos d'ús

##### Casos d'ús de validació



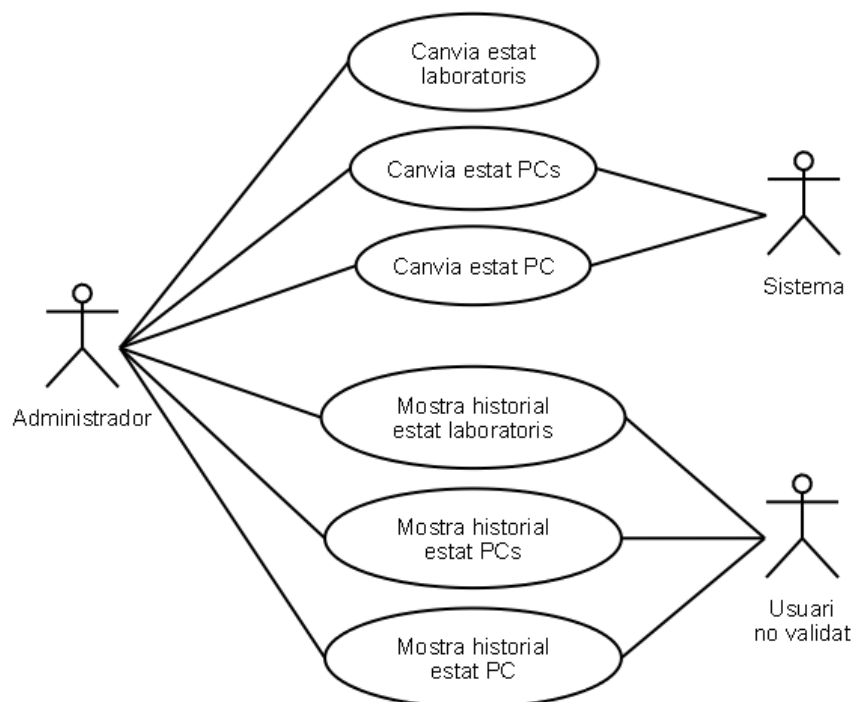
*Figura 4.1.2: Diagrama de casos d'ús de validació*

### Casos d'ús de consulta a l'inventari (de PCs, laboratoris i edificis)

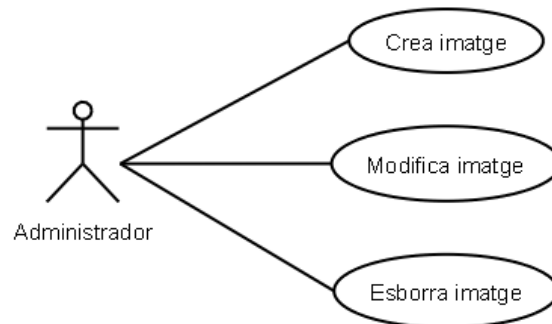
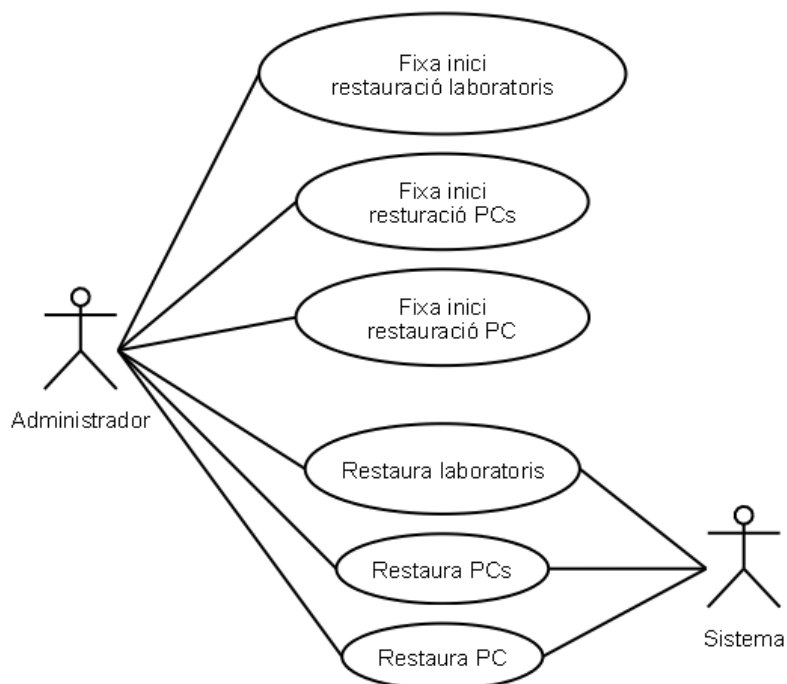


*Figura 4.1.3: Diagrama de casos d'ús de consulta a l'inventari*

### Casos d'ús de gestió d'estats (de PCs i laboratoris)



*Figura 4.1.4: Diagrama de casos d'ús de gestió d'estats*

**Casos d'ús de gestió d'imatges***Figura 4.1.5: Diagrama de casos d'ús de gestió d'imatges***Casos d'ús de manteniment (de PCs i laboratoris)***Figura 4.1.6: Diagrama de casos d'ús de manteniment*

#### 4.1.1.3. Especificació dels casos d'ús

##### Casos d'ús de validació

<b>Cas d'ús:</b>	Valida	
<b>Actors:</b>	Usuari no validat	
<b>Propòsit:</b>	Deixar passar l'actor, o no, a una zona restringida de l'aplicació.	
<b>Resum:</b>	Quan un actor intenta accedir a una zona restringida de l'aplicació, el sistema li demana el nom d'usuari i la contrasenya perquè es validi.	
<b>Flux típic d'esdeveniments:</b>	<b>Accions dels actors</b>	<b>Resposta del sistema</b>
	<p>1. L'actor comunica al sistema que vol accedir a una zona restringida.</p> <p>3. L'actor entra el nom d'usuari i la contrasenya.</p>	<p>2. El sistema demana a l'actor que introdueixi el nom d'usuari i la contrasenya.</p> <p>4. El sistema comprova que l'autenticació és correcta i que l'usuari té privilegis per realitzar l'acció demanada.</p> <p>5. El sistema dona l'actor per validat com aUsuari autenticat (o bé Administrador).</p>
<b>Fluxos alternatius:</b>	<p><b>Línia 2:</b> Si l'actor ja estava validat prèviament, el sistema salta directament a la <b>línia 5</b> sense demanar res a l'actor.</p> <p><b>Línia 4:</b> Si el nom d'usuari o la contrasenya són incorrectes, o bé l'actor no té privilegis per realitzar l'acció demanada el sistema mostra un missatge d'error i torna a la <b>línia 2</b>.</p>	

<b>Cas d'ús:</b>	Desconnecta	
<b>Actors:</b>	Administrador	
<b>Propòsit:</b>	Desconnectar l'actor i transformar-lo en Usuari no validat.	
<b>Resum:</b>	Quan un actor vol sortir d'una zona restringida de l'aplicació, li ho comunica al sistema, i aquest el desconnecta.	
<b>Flux típic d'esdeveniments:</b>	<b>Accions dels actors</b>	<b>Resposta del sistema</b>
	1. L'actor comunica al sistema que vol sortir d'una zona restringida.	5. El sistema desconnecta l'actor: el passa a considerar com a Usuari no validat.
<b>Fluxos alternatius:</b>	-	

#### Casos d'ús de consulta a l'inventari (de PCs, laboratoris i edificis)

<b>Cas d'ús:</b>	Mostra edificis	
<b>Actors:</b>	Usuari no validat / Administrador	
<b>Propòsit:</b>	Mostrar la llista de tots els edificis del campus.	
<b>Resum:</b>	L'actor demana al sistema la llista de tots els edificis del campus, i el sistema li mostra aquest llistat demanat.	
<b>Flux típic d'esdeveniments:</b>	<b>Accions dels actors</b>	<b>Resposta del sistema</b>
	1. L'actor comunica al sistema que vol veure tots els edificis del CBL.	2. El sistema mostra a l'actor un llistat de tots els edificis del campus.
<b>Fluxos alternatius:</b>	-	

#### 4. Especificació i disseny

<b>Cas d'ús:</b>	Mostra laboratoris	
<b>Actors:</b>	Usuari no validat / Administrador	
<b>Propòsit:</b>	Mostrar la llista i l'estat de tots els laboratoris d'un edifici.	
<b>Resum:</b>	L'actor selecciona un edifici i el sistema en mostra la llista de tots els laboratoris. Per a cada laboratori de la llista, també en mostra l'estat.	
<b>Flux típic d'esdeveniments:</b>	<b>Accions dels actors</b>	<b>Resposta del sistema</b>
	<ol style="list-style-type: none"> <li>1. L'actor selecciona un edifici.</li> <li>2. L'actor comunica al sistema que vol veure els laboratoris de l'edifici seleccionat.</li> </ol>	<ol style="list-style-type: none"> <li>3. El sistema mostra a l'actor un llistat de tots els laboratoris de l'edifici indicat, i n'ensenyà (per a cada un) l'estat.</li> </ol>
<b>Fluxos alternatius:</b>	-	

<b>Cas d'ús:</b>	Mostra PCs	
<b>Actors:</b>	Usuari no validat / Administrador	
<b>Propòsit:</b>	Mostrar la distribució i l'estat de tots els PCs d'un laboratori.	
<b>Resum:</b>	L'actor selecciona un laboratori d'un edifici i el sistema en mostra la distribució (en forma de mapa) de tots els PCs. A més, dintre de cada casella de PC, en mostra l'estat i un resum de les seves dades.	
<b>Flux típic d'esdeveniments:</b>	<b>Accions dels actors</b>	<b>Resposta del sistema</b>
	<ol style="list-style-type: none"> <li>1. L'actor selecciona un laboratori.</li> <li>2. L'actor comunica al sistema que vol veure els PCs del laboratori seleccionat.</li> </ol>	<ol style="list-style-type: none"> <li>3. El sistema mostra a l'actor la distribució de tots els PCs del laboratori indicat, i n'ensenyà (per a cada un) l'estat i un resum de les seves dades.</li> </ol>
<b>Fluxos alternatius:</b>	-	

<b>Cas d'ús:</b>	Mostra PC	
<b>Actors:</b>	Usuari no validat / Administrador	
<b>Propòsit:</b>	Mostrar les dades i l'estat d'un PC.	
<b>Resum:</b>	L'actor selecciona un PC concret i el sistema en mostra (en forma de fitxa) totes les dades inventariades. A més, també en mostra l'estat.	
<b>Flux típic d'esdeveniments:</b>	<b>Accions dels actors</b>	<b>Resposta del sistema</b>
	<ol style="list-style-type: none"> <li>1. L'actor selecciona un PC.</li> <li>2. L'actor comunica al sistema que vol veure les dades del PC seleccionat.</li> </ol>	<ol style="list-style-type: none"> <li>3. El sistema mostra a l'actor totes les dades del PC indicat, i n'ensenyà l'estat.</li> </ol>
<b>Fluxos alternatius:</b>	-	

### Casos d'ús de gestió d'estats (de PCs i laboratoris)

<b>Cas d'ús:</b>	Modifica estat laboratoris	
<b>Actors:</b>	Administrador	
<b>Propòsit:</b>	Modificar l'estat dels laboratoris seleccionats.	
<b>Resum:</b>	L'actor escull una sèrie de laboratoris d'un edifici i en canvia l'estat.	
<b>Flux típic d'esdeveniments:</b>	<b>Accions dels actors</b>	<b>Resposta del sistema</b>
	<ol style="list-style-type: none"> <li>1. L'actor selecciona un o més laboratoris d'un edifici.</li> <li>2. L'actor escull l'estat desitjat per als laboratoris seleccionats.</li> <li>3. L'actor comunica al sistema que vol modificar l'estat dels laboratoris seleccionats, pel nou estat escollit.</li> </ol>	<ol style="list-style-type: none"> <li>4. El sistema modifica a la base de dades l'estat dels laboratoris indicats.</li> </ol>
<b>Fluxos alternatius:</b>	-	

#### 4. Especificació i disseny

<b>Cas d'ús:</b>	Modifica estat PCs	
<b>Actors:</b>	Administrador / Sistema	
<b>Propòsit:</b>	Modificar l'estat dels PCs seleccionats.	
<b>Resum:</b>	L'actor escull una sèrie de PCs d'un laboratori (o ja li venen escollits) i en canvia l'estat per un altre.	
<b>Flux típic d'esdeveniments:</b>	<b>Accions dels actors</b>	<b>Resposta del sistema</b>
	<ol style="list-style-type: none"> <li>1. Si l'actor és l'Administrador, selecciona un o més PCs d'un laboratori.</li> <li>2. Si l'actor és l'Administrador, escull l'estat desitjat per als PCs seleccionats.</li> <li>3. L'actor comunica al sistema que vol modificar l'estat dels PCs seleccionats, pel nou estat escollit.</li> </ol>	<ol style="list-style-type: none"> <li>4. El sistema modifica a la base de dades l'estat dels PCs indicats.</li> </ol>
<b>Fluxos alternatius:</b>	-	

<b>Cas d'ús:</b>	Modifica estat PC	
<b>Actors:</b>	Administrador / Sistema	
<b>Propòsit:</b>	Modificar l'estat d'un PC concret.	
<b>Resum:</b>	L'actor escull un PC concret (o ja li ve escollit) i en canvia l'estat per un altre.	
<b>Flux típic d'esdeveniments:</b>	<b>Accions dels actors</b>	<b>Resposta del sistema</b>
	<ol style="list-style-type: none"> <li>1. Si l'actor és l'Administrador, selecciona un PC.</li> <li>2. Si l'actor és l'Administrador, escull l'estat desitjat per al PC seleccionats.</li> <li>3. L'actor comunica al sistema que vol modificar l'estat del PC seleccionat, pel nou estat escollit.</li> </ol>	<ol style="list-style-type: none"> <li>4. El sistema modifica a la base de dades l'estat dels PC indicat.</li> </ol>
<b>Fluxos alternatius:</b>	-	



<b>Cas d'ús:</b>	Mostra historial estat laboratoris	
<b>Actors:</b>	Usuari no validat / Administrador	
<b>Propòsit:</b>	Mostrar l'historial dels estats dels laboratoris seleccionats.	
<b>Resum:</b>	L'actor selecciona una sèrie de laboratoris d'un edifici i el sistema en mostra l'historial dels seus estats.	
<b>Flux típic d'esdeveniments:</b>	<b>Accions dels actors</b>	<b>Resposta del sistema</b>
	<ol style="list-style-type: none"> <li>1. L'actor selecciona un o més laboratoris d'un edifici.</li> <li>2. L'actor comunica al sistema que vol veure l'historial dels estats dels laboratoris seleccionats.</li> </ol>	<ol style="list-style-type: none"> <li>3. El sistema mostra a l'actor l'historial d'estats dels laboratoris indicats per aquest.</li> </ol>
<b>Fluxos alternatius:</b>	-	

<b>Cas d'ús:</b>	Mostra historial estat PCs	
<b>Actors:</b>	Usuari no validat / Administrador	
<b>Propòsit:</b>	Mostrar l'historial dels estats dels PCs seleccionats.	
<b>Resum:</b>	L'actor selecciona una sèrie de PCs d'un laboratori i el sistema en mostra l'historial dels seus estats.	
<b>Flux típic d'esdeveniments:</b>	<b>Accions dels actors</b>	<b>Resposta del sistema</b>
	<ol style="list-style-type: none"> <li>1. L'actor selecciona un o més PCs d'un laboratori.</li> <li>2. L'actor comunica al sistema que vol veure l'historial dels estats dels PCs seleccionats.</li> </ol>	<ol style="list-style-type: none"> <li>3. El sistema mostra a l'actor l'historial d'estats dels PCs indicats per aquest.</li> </ol>
<b>Fluxos alternatius:</b>	-	

#### 4. Especificació i disseny

<b>Cas d'ús:</b>	Mostra historial estat PC	
<b>Actors:</b>	Usuari no validat / Administrador	
<b>Propòsit:</b>	Mostrar l'historial dels estats d'un PC concret.	
<b>Resum:</b>	L'actor selecciona un PC concret i el sistema en mostra l'historial dels seus estats.	
<b>Flux típic d'esdeveniments:</b>	<b>Accions dels actors</b>	<b>Resposta del sistema</b>
	1. L'actor selecciona un PC. 2. L'actor comunica al sistema que vol veure l'historial dels estats del PC seleccionat.	3. El sistema mostra a l'actor l'historial d'estats del PC indicat per aquest.
<b>Fluxos alternatius:</b>	-	

#### Casos d'ús de gestió d'imatges

<b>Cas d'ús:</b>	Crea imatge	
<b>Actors:</b>	Administrador	
<b>Propòsit:</b>	Afegir una imatge al sistema.	
<b>Resum:</b>	L'actor crea una imatge (que s'afegeix al sistema) omplint totes les dades obligatòries.	
<b>Flux típic d'esdeveniments:</b>	<b>Accions dels actors</b>	<b>Resposta del sistema</b>
	1. L'actor entra les dades de la nova imatge en un formulari preparat per a això. 2. L'actor comunica al sistema que vol crear una nova imatge.	3. El sistema afegeix aquesta nova imatge a la base de dades.
<b>Fluxos alternatius:</b>	-	

<b>Cas d'ús:</b>	Modifica imatge	
<b>Actors:</b>	Administrador	
<b>Propòsit:</b>	Modifica una imatge del sistema.	
<b>Resum:</b>	L'actor escull una imatge i la modifica.	
<b>Flux típic d'esdeveniments:</b>	<b>Accions dels actors</b>	<b>Resposta del sistema</b>
	1. L'actor escull una imatge. 2. L'actor comunica al sistema que vol modificar la imatge escollida.  4. L'actor modifica les dades del formulari. 5. L'actor confirma la modificació.	3. El sistema mostra un formulari a l'actor amb les dades de la imatge perquè aquest les modifiqui.  6. El sistema desa aquests canvis de la imatge a la base de dades.
<b>Fluxos alternatius:</b>	-	

<b>Cas d'ús:</b>	Esborra imatge	
<b>Actors:</b>	Administrador	
<b>Propòsit:</b>	Elimina una imatge del sistema.	
<b>Resum:</b>	L'actor escull una imatge i el sistema l'elimina.	
<b>Flux típic d'esdeveniments:</b>	<b>Accions dels actors</b>	<b>Resposta del sistema</b>
	1. L'actor escull una imatge. 2. L'actor comunica al sistema que vol esborrar la imatge escollida.  4. L'actor confirma que vol eliminar la imatge escollida.	3. El sistema mostra un formulari de confirmació.  5. El sistema esborra la imatge de la base de dades.
<b>Fluxos alternatius:</b>	-	

### Casos d'ús de manteniment (de PCs i laboratoris)

<b>Cas d'ús:</b>	Fixa inici restauració laboratoris	
<b>Actors:</b>	Administrador	
<b>Propòsit:</b>	Fixar un moment futur perquè s'iniciï la restauració de laboratoris.	
<b>Resum:</b>	L'actor escull un o més laboratoris i introdueix una data i una hora en la que s'iniciarà la restauració d'aquests; i un període per repetir-la.	
<b>Flux típic d'esdeveniments:</b>	<b>Accions dels actors</b>	<b>Resposta del sistema</b>
	<ol style="list-style-type: none"> <li>1. L'actor selecciona un o més laboratoris d'un edifici.</li> <li>2. L'actor entra les dades de quan ha començar la restauració i amb quina periodicitat s'ha de fer, en un formulari preparat per a això.</li> <li>3. L'actor comunica al sistema que vol fixar l'inici de la restauració dels laboratoris escollits.</li> </ol>	<ol style="list-style-type: none"> <li>4. El sistema programa la restauració dels laboratoris indicats.</li> </ol>
<b>Fluxos alternatius:</b>	-	

<b>Cas d'ús:</b>	Fixa inici restauració PCs	
<b>Actors:</b>	Administrador	
<b>Propòsit:</b>	Fixar un moment futur perquè s'iniciï la restauració dels PCs seleccionats.	
<b>Resum:</b>	L'actor escull una sèrie de PCs d'un laboratori i introdueix una data i una hora en la que s'iniciarà la restauració d'aquests. També entra un període per anar-la repetint regularment.	
<b>Flux típic d'esdeveniments:</b>	<b>Accions dels actors</b>	<b>Resposta del sistema</b>
	<ol style="list-style-type: none"> <li>1. L'actor escull un o més PCs.</li> <li>2. L'actor entra les dades de quan ha començar la restauració i amb quina periodicitat s'ha de fer, en un formulari preparat per a això.</li> <li>3. L'actor comunica al sistema que vol fixar l'inici de la restauració dels PCs escollits.</li> </ol>	<ol style="list-style-type: none"> <li>4. El sistema programa la restauració dels PCs indicats.</li> </ol>
<b>Fluxos alternatius:</b>	-	

<b>Cas d'ús:</b>	Fixa inici restauració PC	
<b>Actors:</b>	Administrador	
<b>Propòsit:</b>	Fixar un moment futur perquè s'iniciï la restauració d'un PC concret.	
<b>Resum:</b>	L'actor escull un PC i introdueix una data i una hora en la que s'iniciarà la restauració d'aquest. També, introdueix un període per repetir-la.	
<b>Flux típic d'esdeveniments:</b>	<b>Accions dels actors</b>	<b>Resposta del sistema</b>
	<ol style="list-style-type: none"> <li>1. L'actor selecciona una o més imatges d'un PC concret.</li> <li>2. L'actor entra les dades de quan ha començar la restauració i amb quina periodicitat s'ha de fer, en un formulari preparat per a això.</li> <li>3. L'actor comunica al sistema que vol fixar l'inici de la restauració de les imatges seleccionades.</li> </ol>	<ol style="list-style-type: none"> <li>4. El sistema programa la restauració de les imatges indicades del PC en qüestió</li> </ol>
<b>Fluxos alternatius:</b>	-	

#### 4. Especificació i disseny

<b>Cas d'ús:</b>	Restaura laboratoris	
<b>Actors:</b>	Administrador / Sistema	
<b>Propòsit:</b>	Restaurar els PCs dels laboratoris seleccionats de manera interactiva o amb el moment d'inici programat.	
<b>Resum:</b>	L'actor escull un o més laboratoris (o ja li venen escollits) i el sistema restaura tots els PCs d'aquests.	
<b>Flux típic d'esdeveniments:</b>	<b>Accions dels actors</b>	<b>Resposta del sistema</b>
	<ol style="list-style-type: none"><li>1. Si l'actor és l'Administrador, selecciona un o més laboratoris d'un edifici.</li><li>2. L'actor comunica al sistema que vol restaurar els PCs dels laboratoris escollits.</li></ol>	<ol style="list-style-type: none"><li>3. El sistema restaura els PCs dels laboratoris seleccionats.</li><li>4. El sistema usa 'Modificar estat PCs' per desar com ha anat la restauració.</li></ol>
<b>Fluxos alternatius:</b>	-	

<b>Cas d'ús:</b>	Restaura PCs	
<b>Actors:</b>	Administrador / Sistema	
<b>Propòsit:</b>	Restaurar els PCs seleccionats de manera interactiva o amb el moment d'inici programat.	
<b>Resum:</b>	L'actor escull un o més PCs d'un laboratori (o ja li venen escollits) i el sistema els restaura tots.	
<b>Flux típic d'esdeveniments:</b>	<b>Accions dels actors</b>	<b>Resposta del sistema</b>
	<ol style="list-style-type: none"><li>1. Si l'actor és l'Administrador, selecciona un o més PCs d'un laboratori.</li><li>2. L'actor comunica al sistema que vol restaurar els PCs escollits.</li></ol>	<ol style="list-style-type: none"><li>3. El sistema restaura els PCs seleccionats.</li><li>4. El sistema usa 'Modificar estat PCs' per desar com ha anat la restauració.</li></ol>
<b>Fluxos alternatius:</b>	-	

<b>Cas d'ús:</b>	Restaura PC	
<b>Actors:</b>	Administrador / Sistema	
<b>Propòsit:</b>	Restaurar un PC concret de manera interactiva o amb el moment d'inici programat.	
<b>Resum:</b>	L'actor escull un PC concret (o ja li ve escollit) i el sistema el restaura.	
<b>Flux típic d'esdeveniments:</b>	<b>Accions dels actors</b> 1. Si l'actor és l'Administrador, selecciona una o més imatges d'un PC concret. 2. L'actor comunica al sistema que vol restaurar les imatges seleccionades del PC escollit.	<b>Resposta del sistema</b> 3. El sistema restaura les imatges indicades del PC en qüestió. 4. El sistema usa 'Modificar estat PC' per desar com ha anat la restauració.
<b>Fluxos alternatius:</b>	-	

## 4.2. Disseny

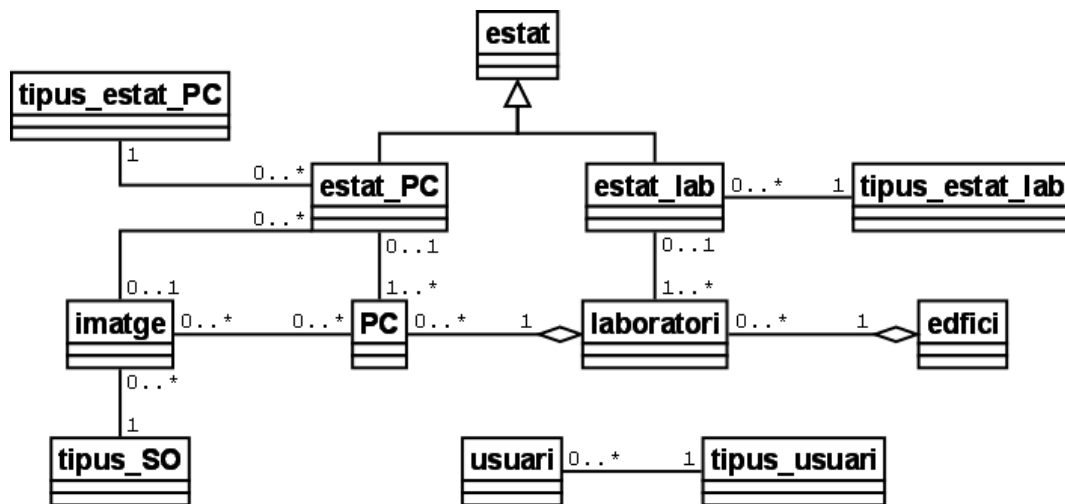
En aquest capítol presentem diverses visions del sistema que ajuden a modelar el seu comportament:

- **Diagrama de classes:** Identifica les classes dels objectes i les seves relacions/associacions. Esdevé una eina molt útil per obtenir el disseny lògic de la bases de dades del sistema.
- **Disseny lògic de la base de dades:** A partir del diagrama de classes, i afegint-li la informació dels atributs i les claus, fem el disseny lògic de la bases de dades del sistema.
- **Diagrames d'activitats:** Ens donen un nou enfocament que ens serà útil en la implementació per guiar el disseny de la interfície amb l'usuari.
- **Arquitectura del *software*:** L'aplicació de diversos patrons de disseny del *software* ens permetrà obtenir un nou diagrama de classes que serà el que implementarem.

- **Diagrames de seqüència dels casos d'ús:** Ens permetran explicar com funcionen cadascun dels casos d'ús. A més, com que ja estaran adaptats al nou diagrama de classes, seran molt útils per a la implementació.

##### 4.2.1. Diagrama de classes

Amb aquest diagrama modelem la informació del sistema. Ens permet identificar les entitats, per tant és bastant equivalent a un diagrama entitat-relació, que ja no dibuixarem.

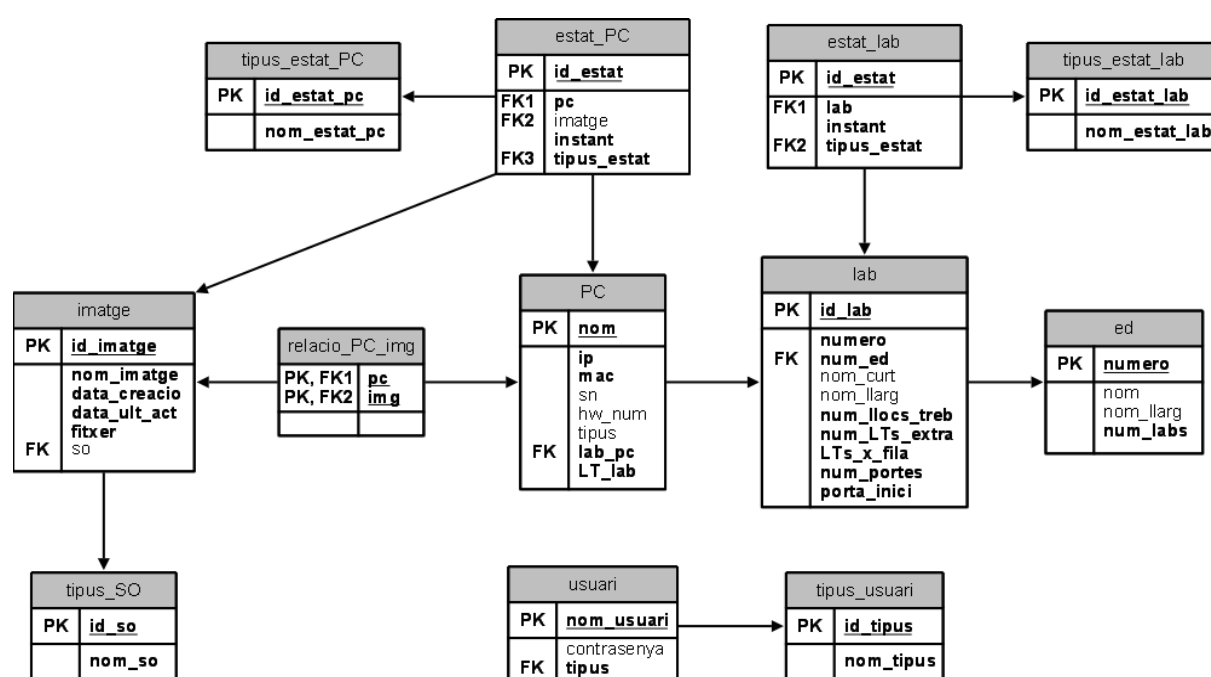


*Figura 4.2.1: Diagrama de classes*



### 4.2.2. Disseny lògic de la base de dades

La base de dades tindrà les entitats que apareixen a la figura *Figura 4.2.2*, amb els atributs i les claus que hem dibuixat.



*Figura 4.2.2: Diagrama lògic de la base de dades*

### 4.2.3. Diagrames d'activitats

Els diagrames d'activitats són un cas especial dels diagrames d'estat en que la majoria d'estats són estats d'acció (identifiquen una acció que es realitza). Per a una aplicació web com la nostra, aquest diagrama modelitza molt bé les accions o activitats que es realitzen a partir de les interaccions de l'usuari amb el sistema.

#### 4. Especificació i disseny

---

Així doncs, aquest diagrama ens pot mostrar les transicions entre les diverses pantalles del sistema, o bé ens pot mostrar com s'encadenen els casos d'ús. Com que l'usuari pot interaccionar lliurement amb el sistema, les transicions mostrades no han de ser sempre certes. Tot i així, els estats representen les principals accions o activitats i les transicions la cadena lògica d'esdeveniments del sistema.

Interpretat com a pantalles, el diagrama ens mostra els canvis de pantalla que es realitzen amb la navegació i ens resultarà molt útil per implementar les transicions entre pantalles.

Per afavorir la lectura del diagrama, l'hem dividit en dos: dibuixem un diagrama d'activitats per als usuaris no validats (que mostra totes les activitats que pot fer un usuari no validat, inclosa la de validar-se) i un altre per als usuaris ja validats (que, lògicament, mostra totes les activitats que pot fer un usuari validat, inclosa la de desconnectar-se). Les línies discontinúes simbolitzen transicions cap a accions que es realitzen fora de la pantalla principal del navegador com, per exemple, l'obertura de noves finestres.

Com a diagrama de pantalles, s'ha de dir que no hi trobarem les pantalles informatives o les d'error, on l'única acció que es realitza és informar a l'usuari. Obviant aquestes pantalles també es fa més llegible el diagrama.

#### 4.2.3.1. Diagrama d'activitats per a usuaris no validats

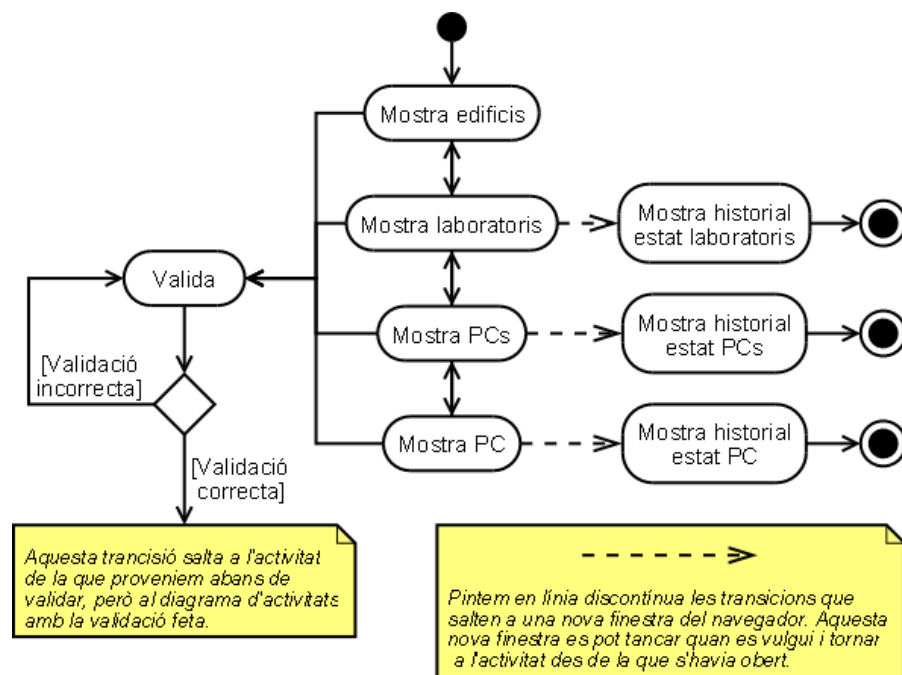


Figura 4.2.3: Diagrama d'activitats per a usuaris no validats

#### 4.2.3.2. Diagrama d'activitats per a usuaris ja validats

És possible que el lector d'aquest treball tingui la sensació de que manca alguna activitat d'introducció de dades abans d'executar algunes accions de l'aplicació com crear imatges, canviar estats i fixar l'inici de les restauracions. Però això només és una percepció perquè a l'hora de mostrar laboratoris o PCs, o bé a l'hora d'executar l'acció Gestiona imatges, ja hem previst que l'usuari de l'aplicació pot voler executar les activitats anteriorment comentades i hem introduït a la pantalla tots els camps necessaris perquè quan l'usuari es decideixi per una acció, la pugui executar amb els mínims passos possibles.

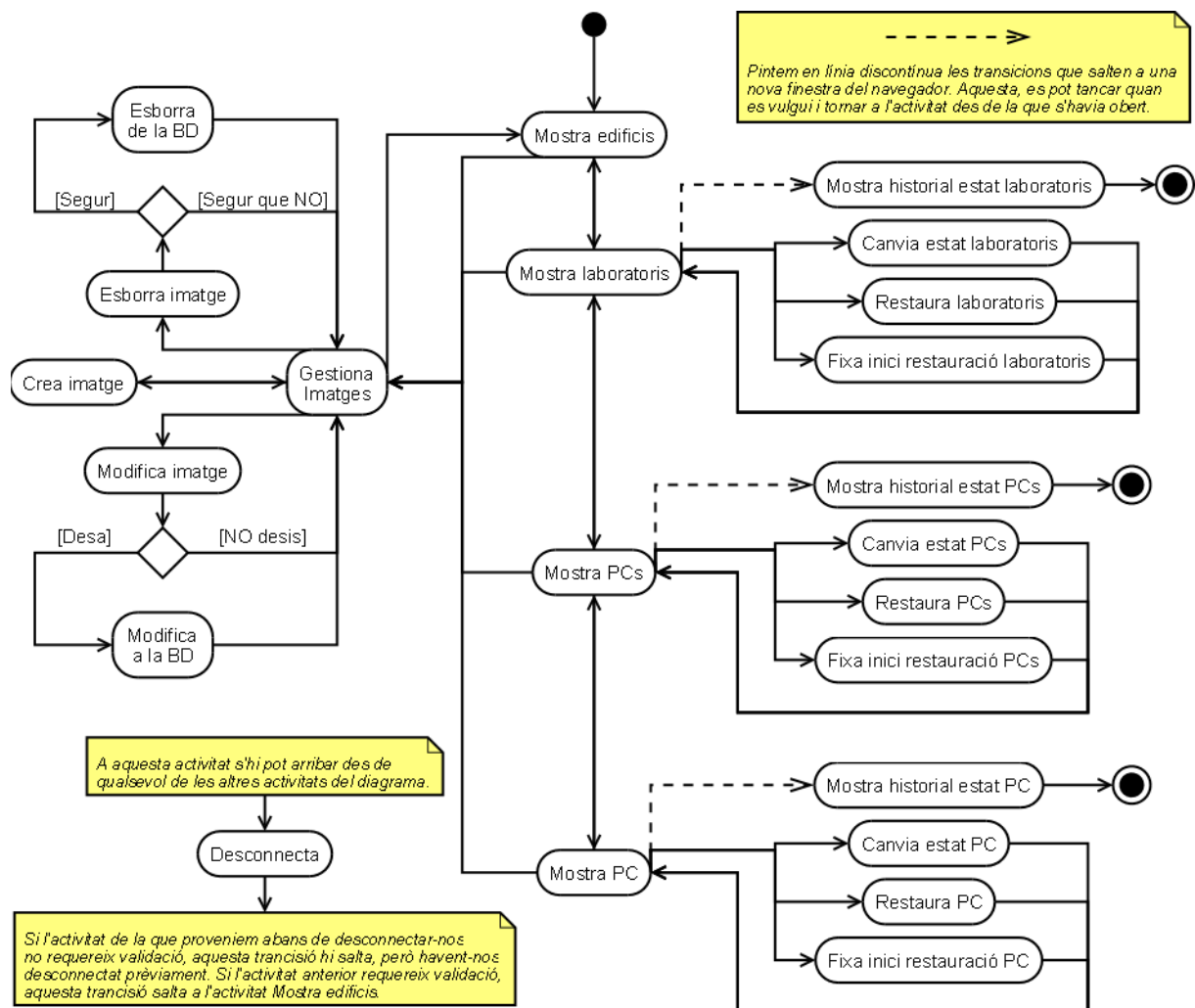


Figura 4.2.4: Diagrama d'activitats per a usuaris ja validats

## 4.2.4. Arquitectura del software

### 4.2.4.1. Patrons arquitectònics i de disseny

Els patrons de disseny expressen esquemes d'organització estructural de sistemes de *software*. En un àmbit més abstracte, els patrons arquitectònics expressen maneres d'organitzar subsistemes o components predefinits que formen part d'un sistema. I en un àmbit més concret, serveixen per a refinar subsistemes i components, o les relacions entre ells.

Aplicant un cert grau d'abstracció, una aplicació web es pot descriure com una sèrie de peticions i recepcions de pàgines web. Llavors, una aplicació web no és gaire diferent d'una pàgina web dinàmica. Una aplicació web és un sistema (servidor web, xarxa, protocol HTTP i navegador) on la interacció (les dades introduïdes i la navegació) afecten l'estat de la lògica de negoci (o el domini de l'aplicació).

Tal i com ja hem vist des del principi d'aquest treball, pel fet de dissenyar un sistema d'informació en un entorn web, ja estem usant un patró arquitectònic client-servidor. Per altra banda, també hem volgut usar un arquitectura en capes que separi en nivells diferenciats, la presentació, el domini i la gestió de les dades:

- **Capa de presentació:** És la que interacciona amb l'usuari. Per tant en l'entorn client-servidor és la que rep les peticions de l'usuari i la que retorna uns resultats a la pantalla del navegador del client.
- **Capa de domini:** És la que engloba tota la lògica del sistema. Per tant és l'encarregada d'assabentar-se, controlar la validesa i executar les accions dels esdeveniments. També és la que obté els resultats de les accions i en comunica les respostes a la capa de presentació.
- **Capa de gestió de dades:** És la que interacciona amb l'SGBD. Executa totes les consultes a la base de dades i en torna els resultats a qui li ha demanat.

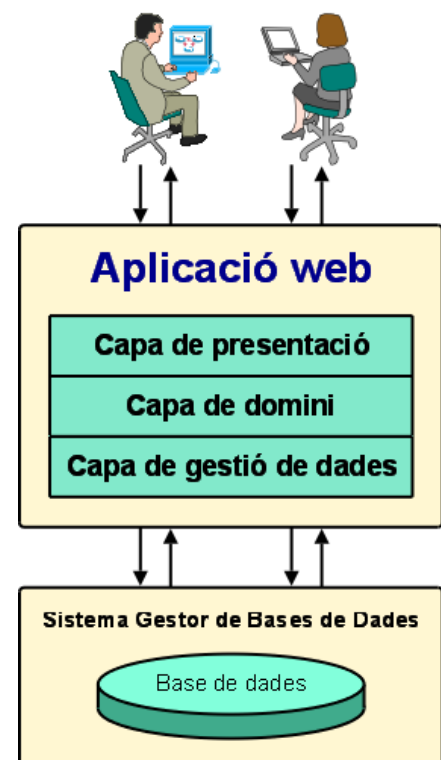


Figura 4.2.5: Arquitectura en 3 capes

##### 4.2.4.2. El patró Model-Vista-Controlador i la seva adaptació per a web

El patró Model-Vista-Controlador (a partir d'ara MVC) és un bon mètode per separar la lògica del domini d'una aplicació de la interfície amb l'usuari. Amb aquesta separació podem fer canvis i afegir noves funcionalitats a una aplicació de forma senzilla.

Amb la separació que s'aconsegueix amb el patró MVC podem fer canvis en la presentació sense que afectin a la capa de domini i viceversa. També es podem tenir diverses vistes per un sol model.

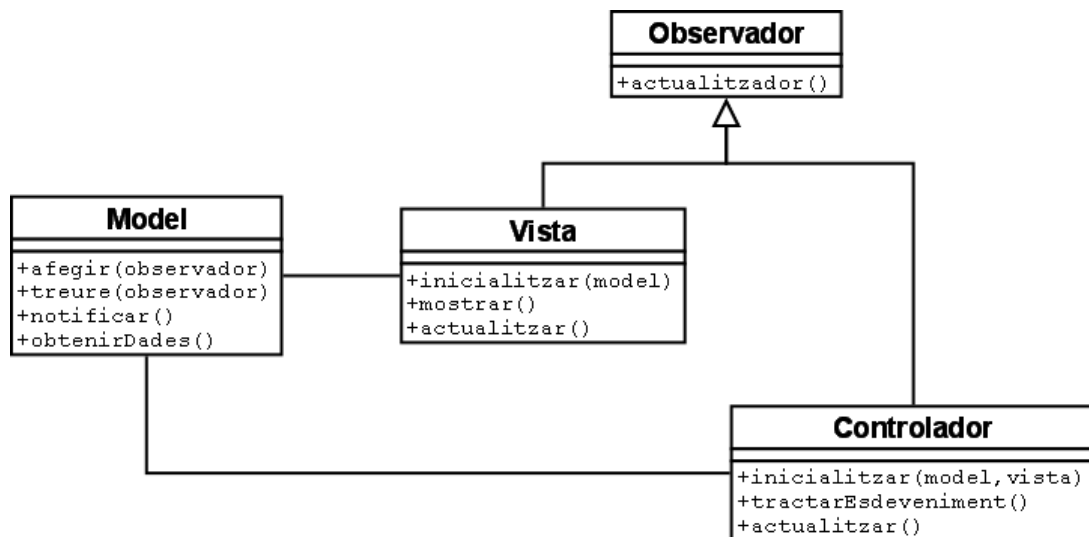


Figura 4.2.6: Patró Model-Vista-Controlador clàssic

Però el patró s'ha d'adaptar al paradigma de la programació web: hi ha diverses adaptacions del patró i moltes d'elles inclouen un marc ja desenvolupat que es pot estendre per a implementar noves aplicacions. Tots aquests marcs són massa complexos per a aplicacions petites i mitjanes.

Per això hem implementat un marc propi, molt més senzill que els anunciats a l'anterior paràgraf, que implementa aquest patró en el paradigma de la programació web.

El patró adaptat divideix els components de l'aplicació en tres parts:

- **Model:** Encapsula la funcionalitat del domini de l'aplicació de tal manera que és independent de la interfície amb l'usuari. No conté referències ni a les vistes ni als controladors, sinó que només serveix dades de forma passiva. A més, en el paradigma

de la programació web optem per un model passiu perquè que no necessitem notificar els canvis en l'estat del model a les vistes ni als controladors. Per tant, podrem prescindir del patró observador que s'acostuma a usar per implementar el patró MVC.

- **Vista:** Obté dades del model i les presenta a l'usuari. No canvia l'estat del model, només el consulta.
- **Controlador:** Rep les entrades de l'aplicació i les tradueix en peticions al model i a la vista. És el que executa operacions sobre el model que canvien el seu estat. Com que usem un model passiu que no notifica canvis en el seu estat, és el propi controlador qui notifica a la vista dels canvis en el model.

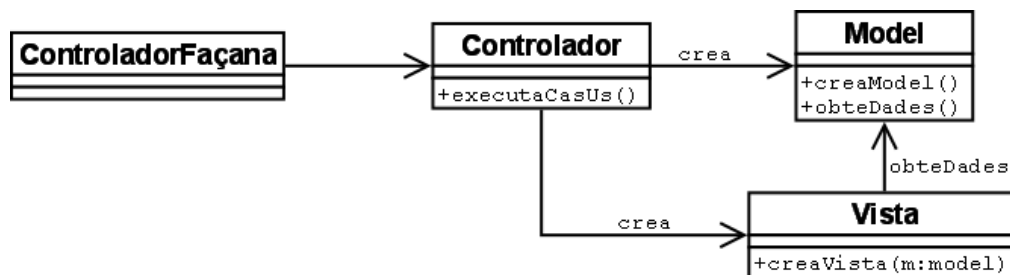


Figura 4.2.7: Patró Model-Vista-Controlador adaptat

#### 4.2.4.3. Controlador façana

El Patró controlador o bé Controlador façana (en anglès *front controller* o també *dispatcher*) s'usa en l'entorn web per centralitzar en un sol punt totes les peticions del sistema.

En l'entorn web, el Controlador façana ens permet centralitzar en un sol punt l'autenticació d'usuaris i les comprovacions de seguretat.

Per al nostre sistema hem optat per un Controlador façana que gestiona totes les peticions al sistema. Aquest controlador gestiona les sessions dels usuaris al sistema (autenticació, permisos, paràmetres, etc) i posteriorment passa les entrades del sistema al controlador que correspongui del patró MVC.

El diagrama de seqüència adjunt (Figura 4.2.8) ens mostra una interacció genèrica de l'usuari amb el SGC. L'usuari únicament interacciona amb un únic objecte que per a ell és la façana del sistema. Però internament, instanciem una classe controlador i executem una operació. Finalment, retornem una sortida a l'usuari a través de l'objecte façana.

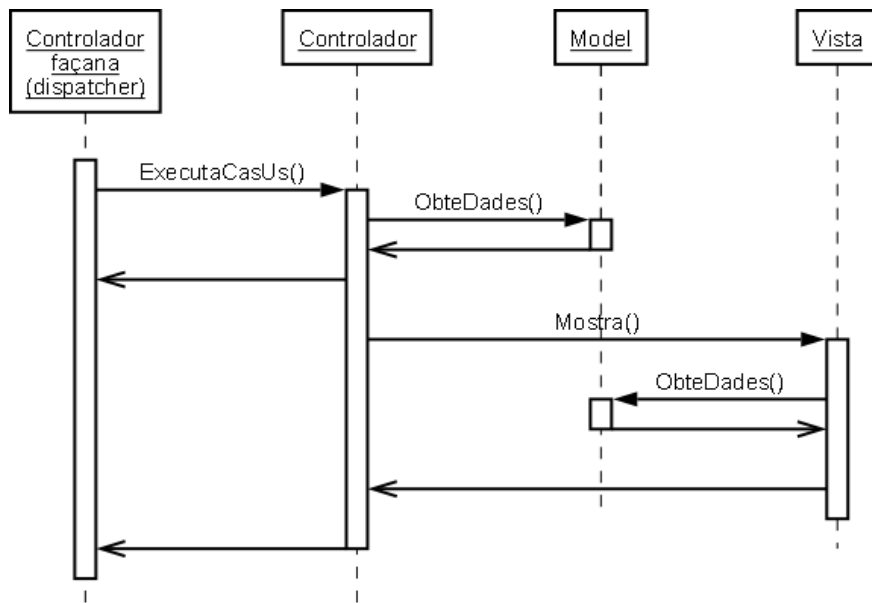


Figura 4.2.8: Diagrama de seqüència del patró MVC adaptat

#### 4.2.5. Diagrama de classes amb el patró MVC

Amb aquest diagrama mostrem les classes corresponents al patró de disseny que hem aplicat. La classe Inici és la que fa la funció de Controlador façana del sistema i centralitza la recepció de les entrades del sistema. Les classes Model, Controlador i Vista, tal i com indiquen els seus noms, corresponen a l'aplicació del patró MVC.

Aquestes 3 classes es poden definir com a abstractes perquè part o tota la seva implementació es fa en una subclasse que conté les operacions necessàries per al cas d'ús que pertoca.

Al diagrama, la separació en capes és molt clara:

- La classe Model i totes les seves subclasses gestionen l'accés a les dades: només es relacionen amb la classe AccésBD que és l'única que realment fa peticions a l'SGBD.
- La classe Controlador i totes classes amb nom Control\_\* executen accions que ni estan relacionades directament amb la gestió de dades, ni amb la presentació. Per exemple envien i reben informació per la xarxa o escriuen i llegeixen fitxers del sistema de fitxers. Totes elles formen part de la capa de domini.
- La classe Vista i totes les seves subclasses formen part de la capa de presentació.



Les classes ControladorFaçana i Controlador estan situades conceptualment entre les classes de la capa de domini i les de la capa presentació. Podem considerar que la classe ControladorFaçana pertany a la capa de presentació perquè interacciona amb l'usuari, tot i que pugui fer altres funcions. Pel que fa a la classe Controlador, no deixa de fer les mateixes funcions que el Controlador façana, però en un segon pla. Per altra banda, també el podem considerar dins la capa de domini perquè té també molta relació amb els models.

En aquest diagrama, ja no hem pintat algunes classes i associacions entre classes que apareixien al diagrama de classes conceptual perquè aquest diagrama ja NO mostra la interacció conceptual entre classes d'objectes o entitats, sinó que posa èmfasi en com es gestiona la interacció del sistema i el seu entorn, i com es desenvolupen els casos d'ús.

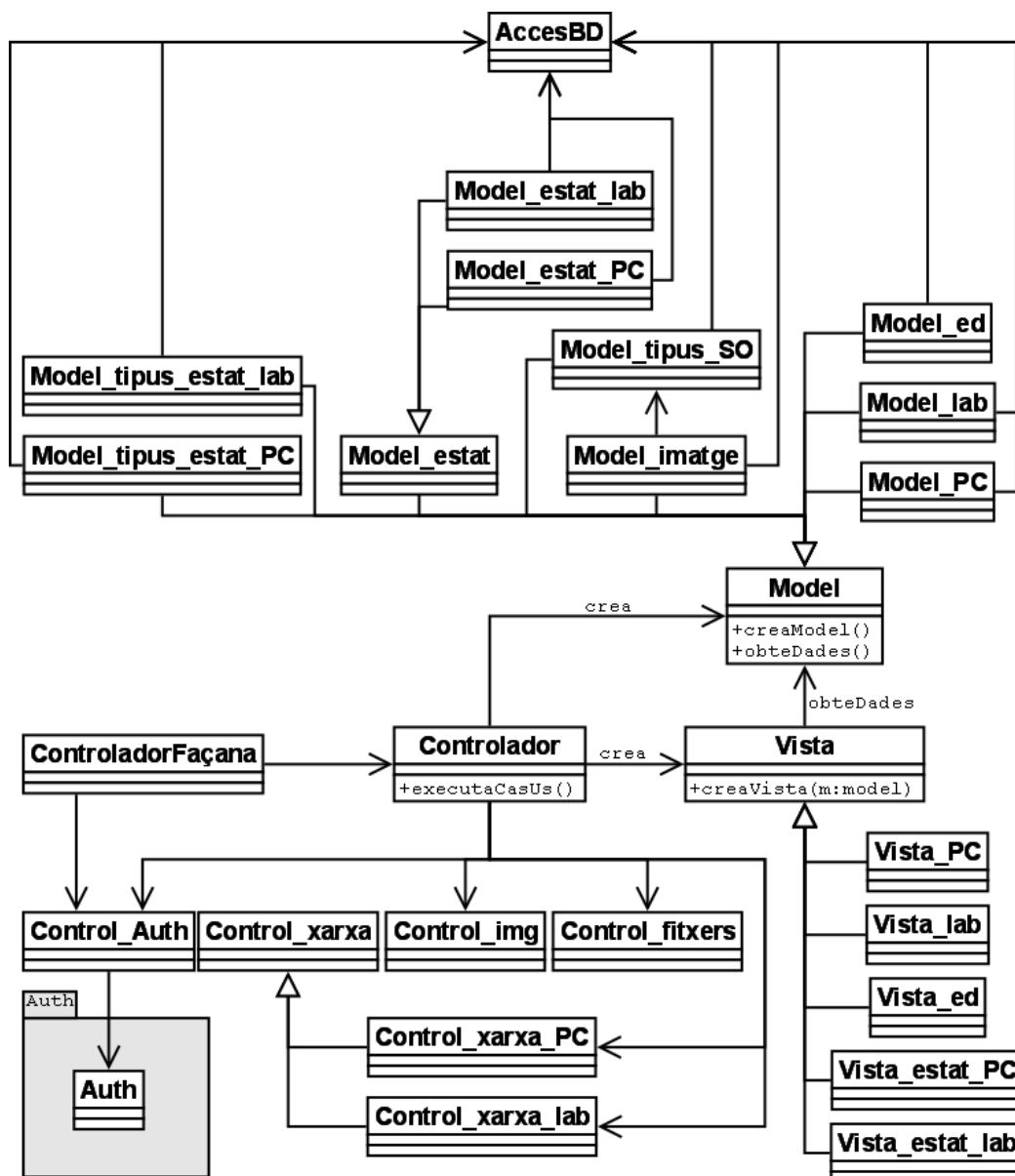


Figura 4.2.9: Diagrama de classes amb el patró MVC

### 4.2.6. Diagrames de seqüència dels casos d'ús

Els diagrames de seqüència dels casos d'ús ens permeten modelar l'escenari on es desenvolupa cada cas. Si en la descripció del cas d'ús, el flux ens mostrava quina funció realitzava el cas, els diagrames de seqüència ens permeten identificar, a alt nivell, com funciona aquest cas i quines operacions hi intervenen.

#### 4.2.6.1. Mostra edificis

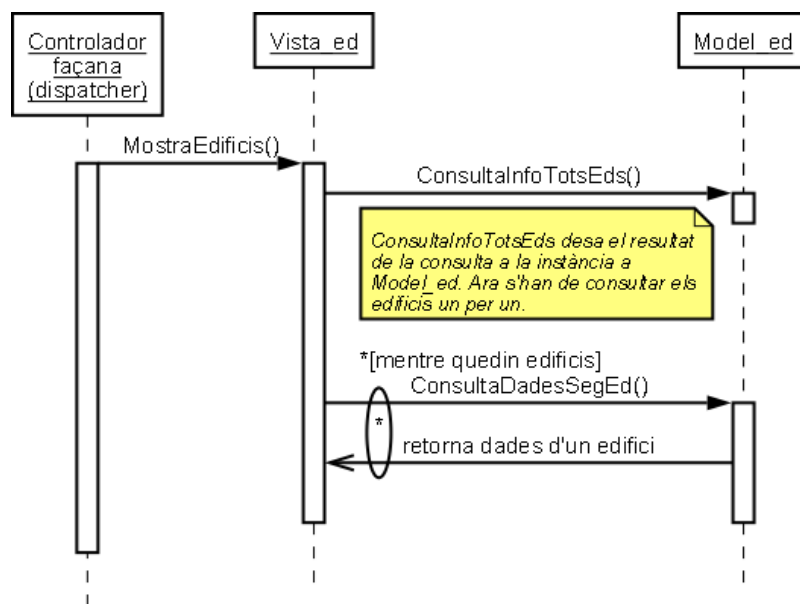


Figura 4.2.10: Diagrama de seqüència del cas d'ús Mostra edificis

## 4.2.6.2. Mostra laboratoris

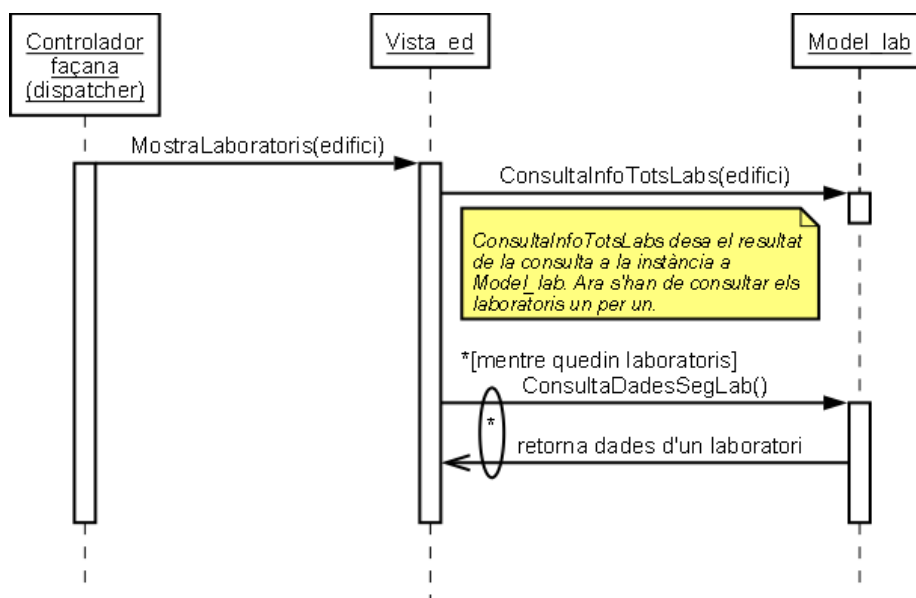


Figura 4.2.11: Diagrama de seqüència del cas d'ús Mostra laboratoris

## 4.2.6.3. Mostra PCs

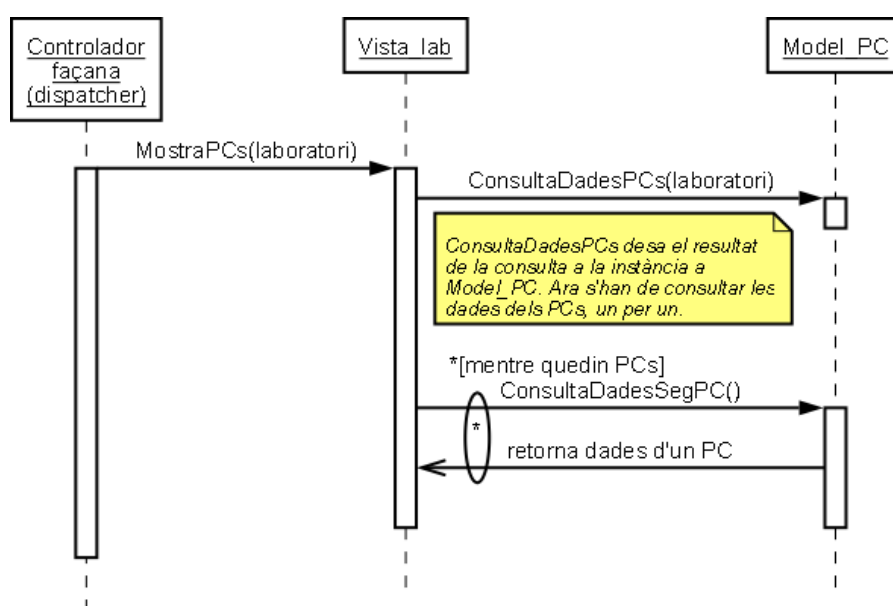


Figura 4.2.12: Diagrama de seqüència del cas d'ús Mostra PCs

#### 4.2.6.4. Mostra PC

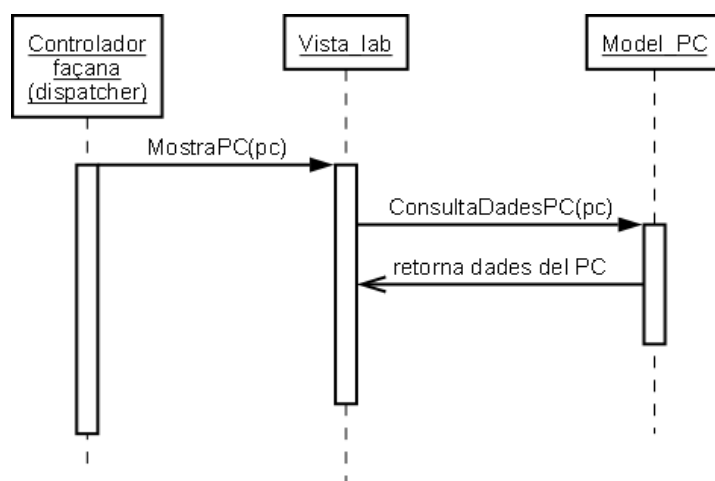


Figura 4.2.13: Diagrama de seqüència del cas d'ús Mostra PC

#### 4.2.6.5. Canvia estat laboratoris

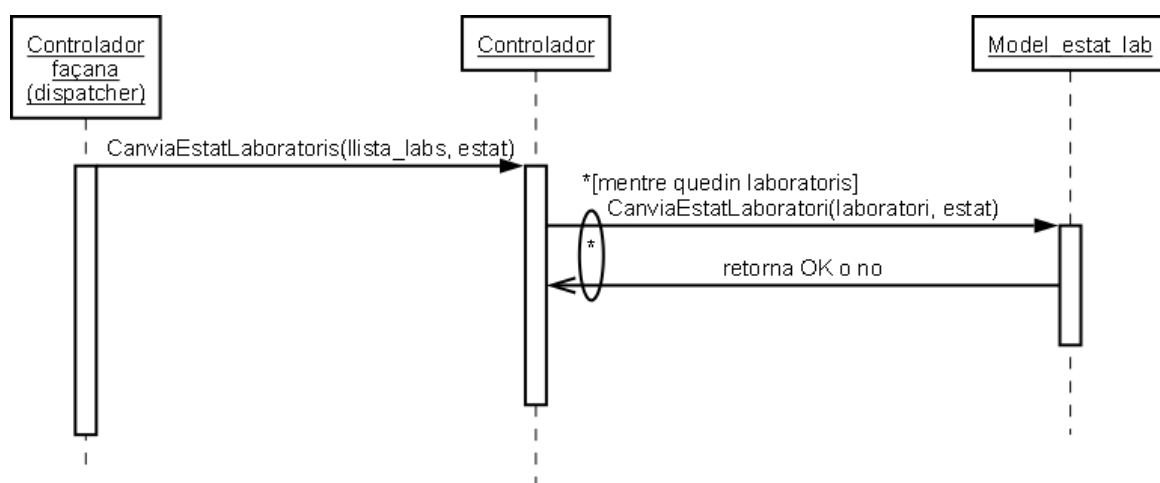


Figura 4.2.14: Diagrama de seqüència del cas d'ús Canvia estat laboratoris

#### 4.2.6.6. Canvia estat PCs

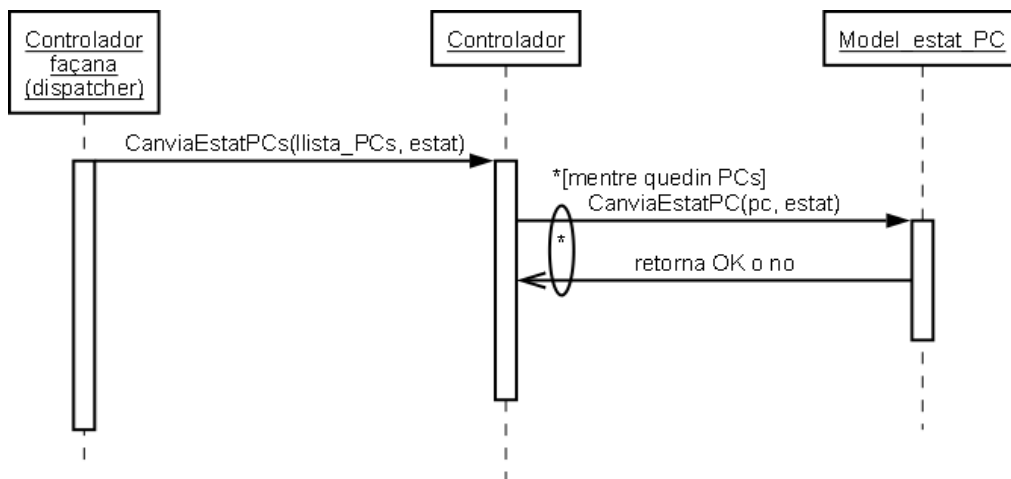


Figura 4.2.15: Diagrama de seqüència del cas d'ús Canvia estat PCs

#### 4.2.6.7. Canvia estat PC

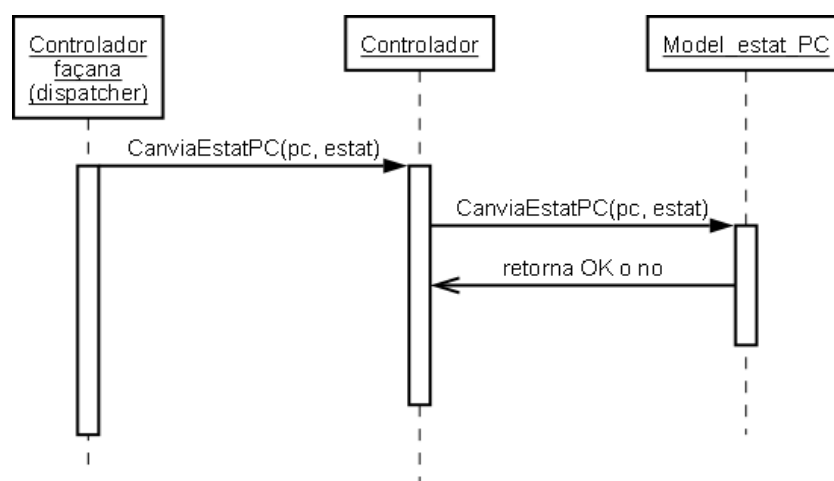


Figura 4.2.16: Diagrama de seqüència del cas d'ús Canvia estat PC

#### 4.2.6.8. Mostra historial estat laboratoris

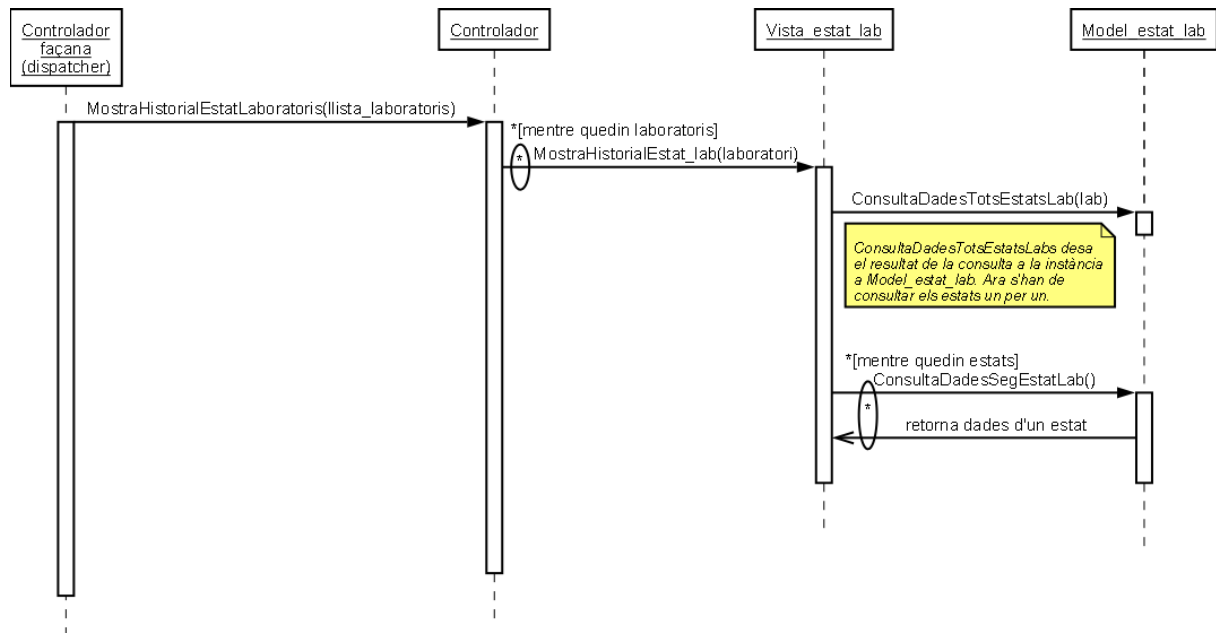


Figura 4.2.17: Diagrama de seqüència del cas d'ús Mostra historial estat laboratoris

#### 4.2.6.9. Mostra historial estat PCs

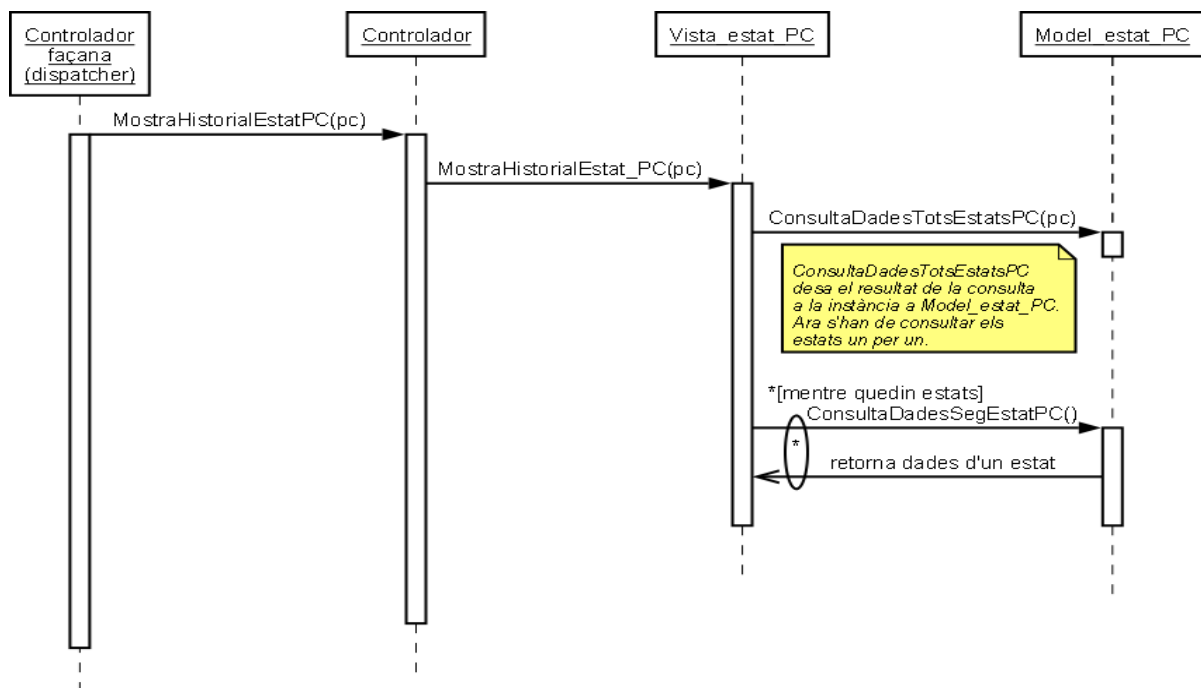


Figura 4.2.18: Diagrama de seqüència del cas d'ús Mostra historial estat PCs

#### 4.2.6.10. Mostra historial estat PC

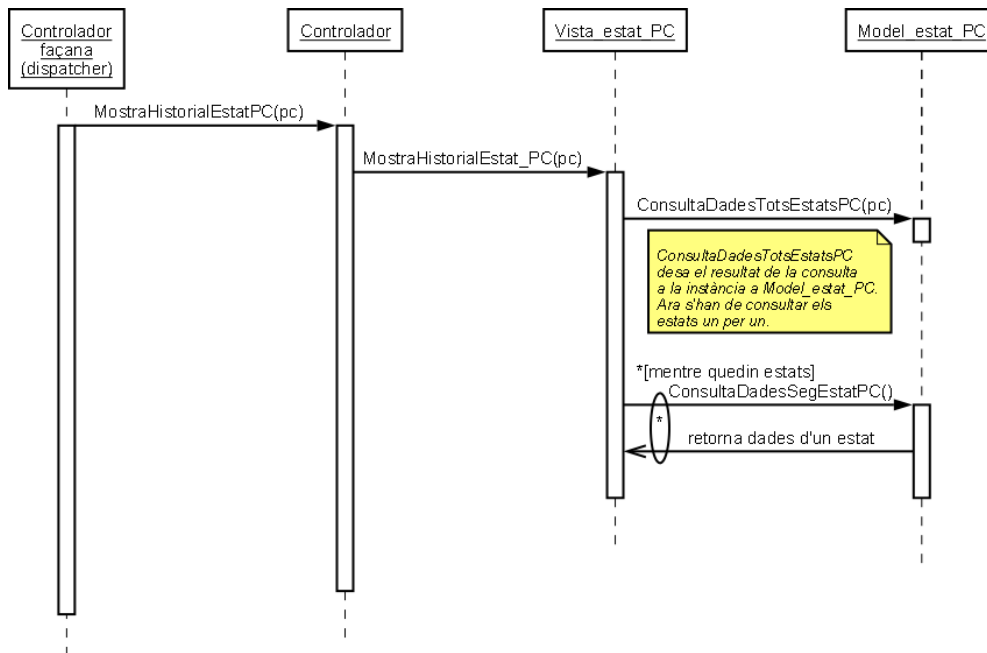


Figura 4.2.19: Diagrama de seqüència del cas d'ús Mostra historial estat PC

#### 4.2.6.11. Crea imatge

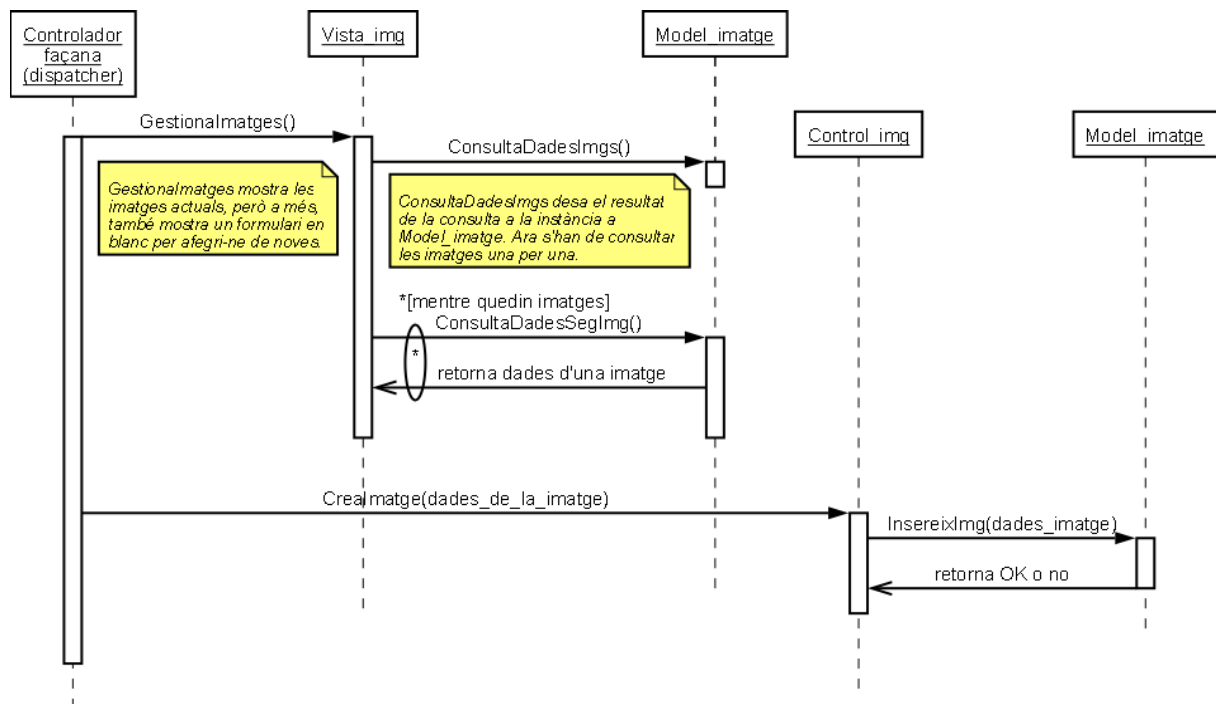


Figura 4.2.20: Diagrama de seqüència del cas d'ús Crea imatge

#### 4.2.6.12. Modifica imatge

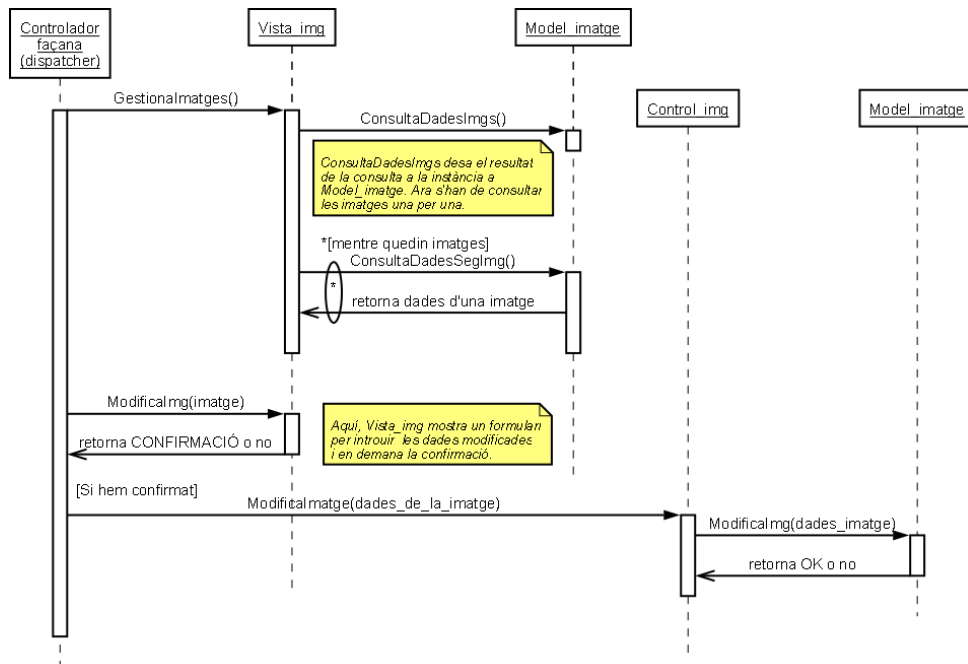


Figura 4.2.21: Diagrama de seqüència del cas d'ús Modifica imatge

#### 4.2.6.13. Esborra imatge

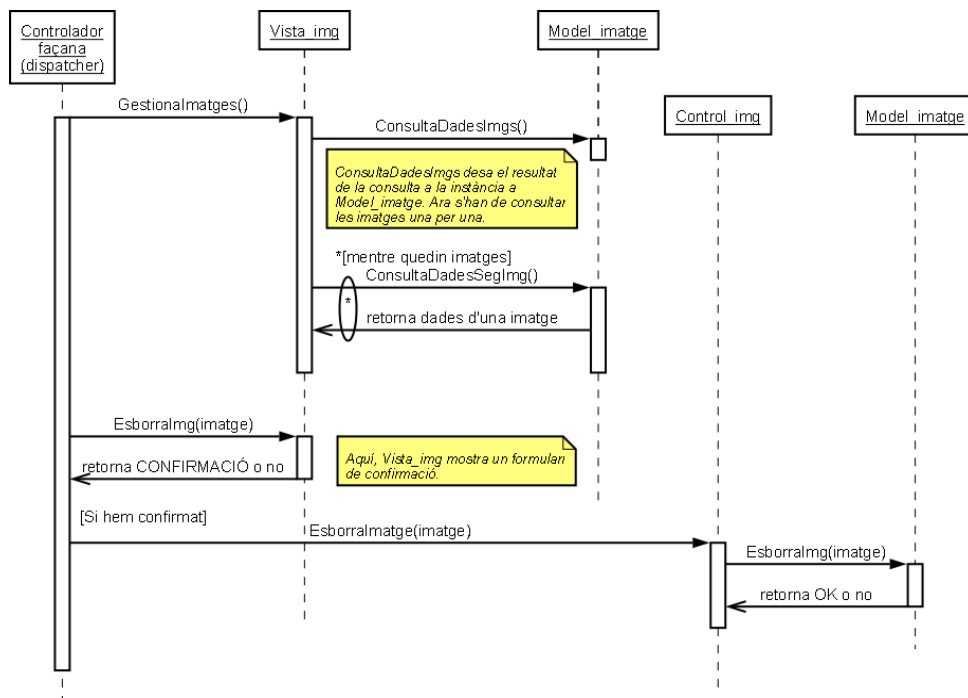


Figura 4.2.22: Diagrama de seqüència del cas d'ús Esborra imatge



## 4.2.6.14. Restaura laboratoris

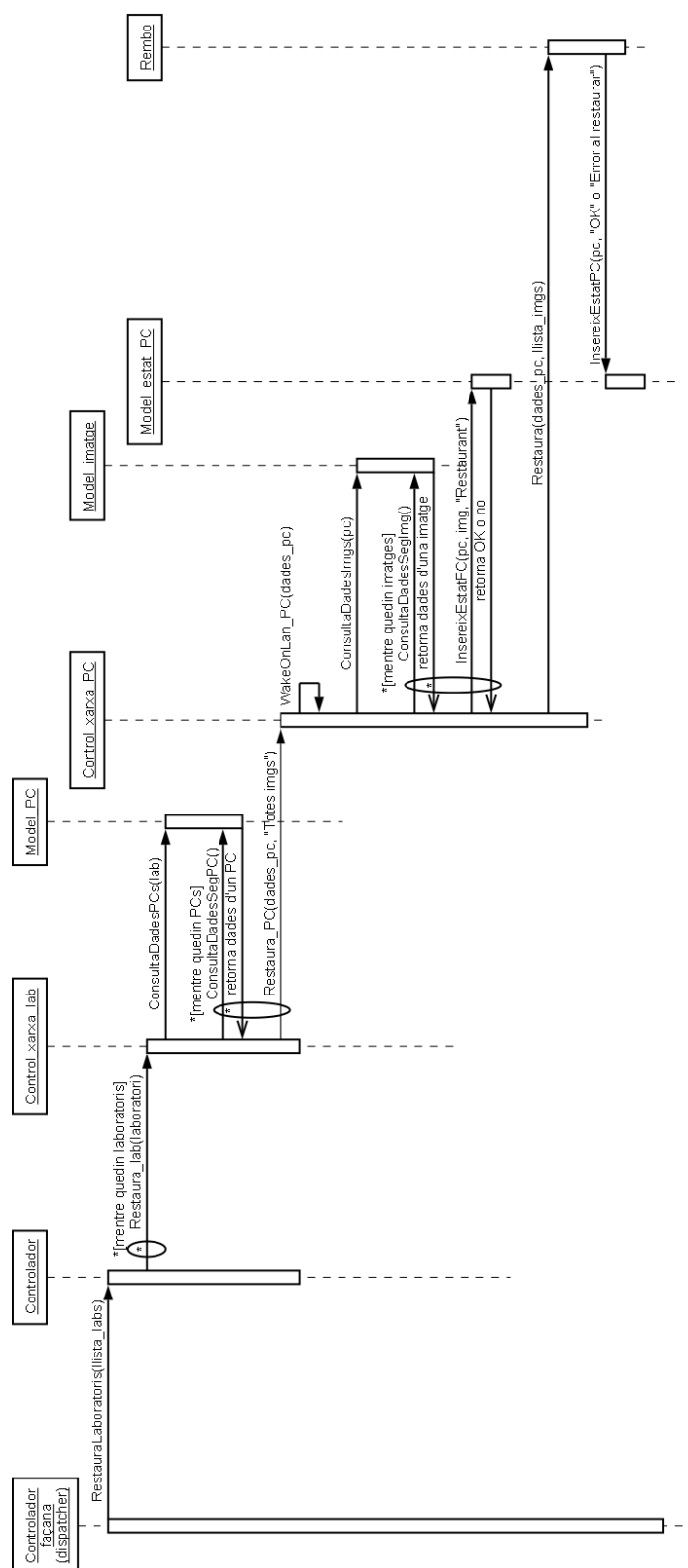


Figura 4.2.23: Diagrama de seqüència del cas d'ús Restaura laboratoris

#### 4.2.6.15. Restaura PCs

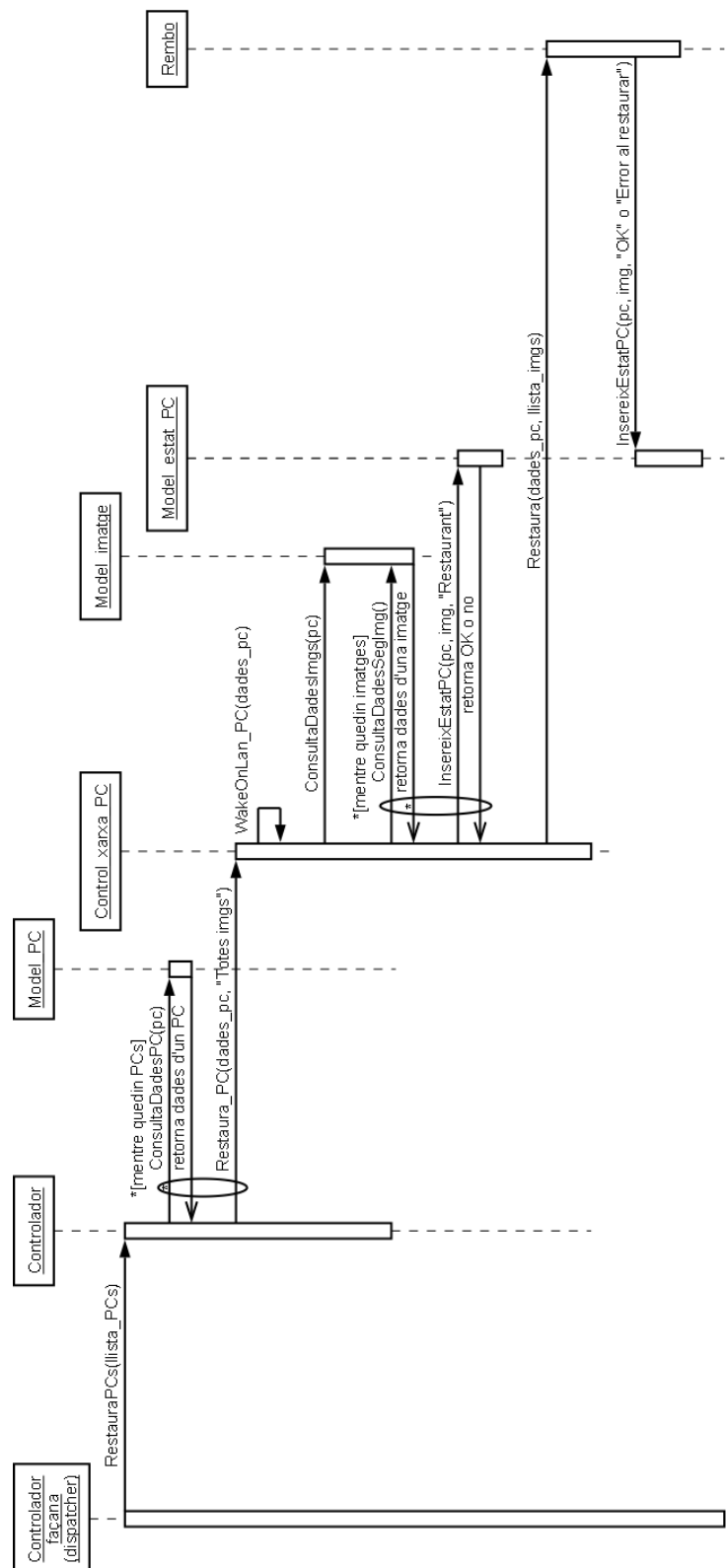


Figura 4.2.24: Diagrama de seqüència del cas d'ús Restaura PCs

## 4.2.6.16. Restaura PC

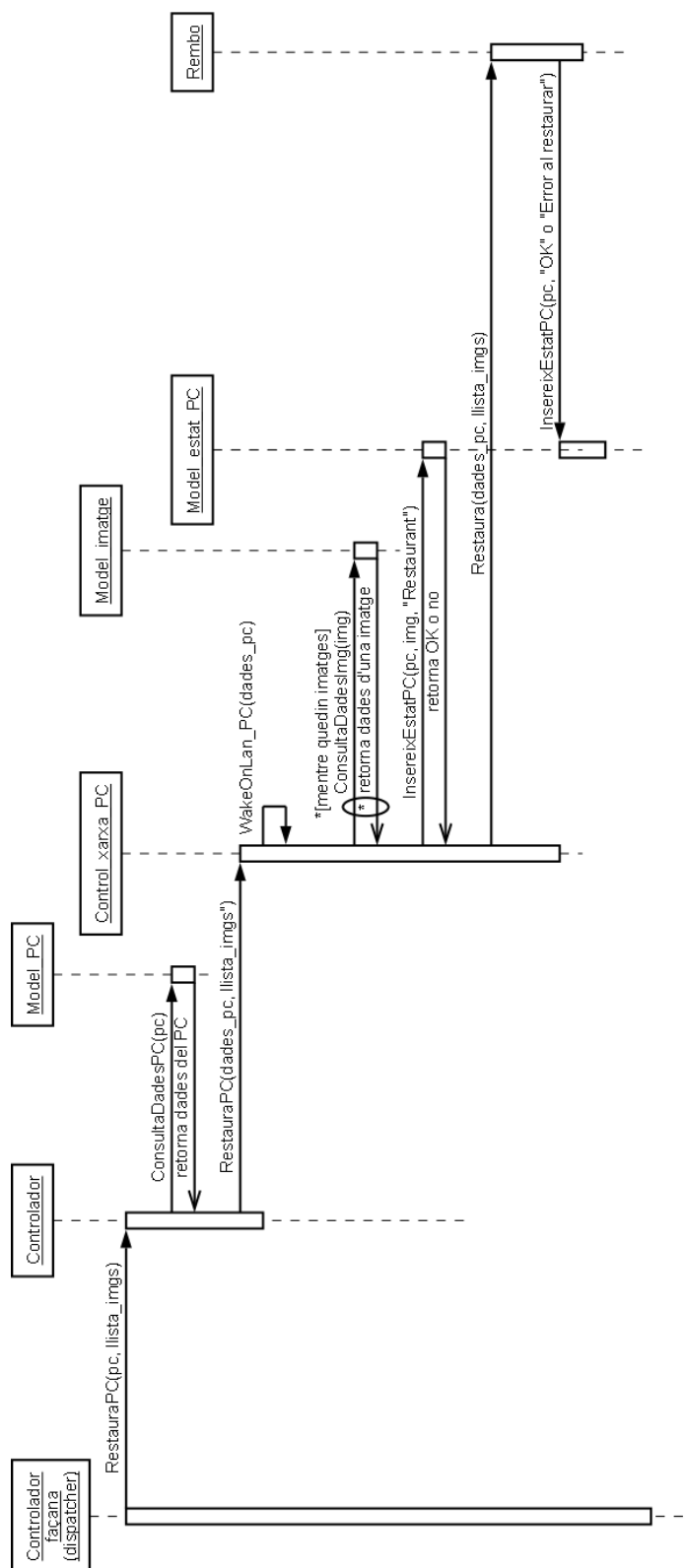


Figura 4.2.25: Diagrama de seqüència del cas d'ús Restaura PC



## 5. Implementació

5.1. Software usat.....	93
5.1.1. Notes sobre PHP.....	94
5.2. Aspectes de seguretat.....	95
5.2.1. Sistema d'autenticació.....	95
5.3. Disseny de la interfície amb l'usuari.....	95
5.4. Documentació.....	98
5.5. Fitxers de l'aplicació.....	98
5.6. Base de dades.....	100
5.6.1. Creació de la base de dades.....	100
5.6.2. Disseny físic de la base de dades.....	100
5.6.3. Primeres insercions.....	103
5.7. Exemples de codi.....	103
5.7.1. index.html: el controlador façana.....	103
5.8. Proves.....	105

En aquest capítol expliquem diverses decisions que hem pres i alguns detalls interessants de saber sobre la implementació.

### 5.1. Software usat

Com que per implementar aquest sistema ha estat necessari instal·lar una sèrie de *software*, hem hagut de decidir sobre quina plataforma l'instal·làvem i quin programari escollíem. Hi havia una cosa que teníem molt clara des del principi d'aquest projecte: volíem usar *software* lliure sempre que fos possible.

Per això, hem instal·lat el sistema operatiu Linux a l'ordinador del que disposàvem per desenvolupar aquest projecte. A més, els servidors del CBL també són Linux i, per tant, tindrem una compatibilitat gairebé total.

L'altre *software* que hem hagut d'instal·lar per implementar el projecte és de codi obert (excepte el Rembo) i és el següent:

- **Apache server (v2.0.55-r1):** Servidor web.
- **PHP (v5.1.6-r8):** Instal·lat com un mòdul de l'Apache.
- **MySQL (v5.0.22):** És l'SGBD.
- **Rembo (v4.0):** *Software* d'imatges. Aquest programa, no és lliure (és *software* propietari i comercial), però l'hem hagut d'usar perquè ens ho requeria el CBL: totes les imatges del CBL estan fetes usant aquest programari.

A més, i tot i que no era estrictament necessari, també hem instal·lat el programa **phpMyAdmin** (v2.7.0-pl2) per poder administrar més fàcilment de la base de dades MySQL implementada al sistema.

Hem editat tots els fitxers i *scripts* de codi amb l'editor **Gedit** de Linux.

Pel disseny de les pàgines web no hem usat cap altre programa i l'hem fet tot a mà. És per això que potser aquest disseny és una mica pobre, però volíem que aquest projecte fos senzill i funcional, i no hem entrat gaire a fons en temes estètics.

### 5.1.1. Notes sobre PHP

Com ja hem anunciat en capítols anteriors, el llenguatge escollit per a la implementació d'aquest sistema ha estat PHP. Aquest llenguatge és un dels més utilitzats per al desenvolupament d'aplicacions web<sup>4</sup> i la majoria de servidors web el suporten. També és un llenguatge de codi obert i la comunitat de desenvolupadors proveeix una documentació molt extensa i moltes llibreries de codi addicional. La versió 5 d'aquest llenguatge (la usada en aquest projecte) té una millora substancial en el suport a la programació orientada objectes tant pel que fa al desenvolupament com en el rendiment. Altres llenguatges que he estudiat han estat, per exemple, Perl i Python. Perl també té una gran difusió, però no tenia suport per a objectes. Per altra banda, Python té una excel·lent orientació a objectes, però no té tanta difusió i té alguns problemes de portabilitat.

El llenguatge PHP funciona sobre la majoria de sistemes operatius de tipus Unix i Windows. El servidor web Apache, l'utilitzat en el desenvolupament, permet la configuració de PHP com a mòdul i així s'aconsegueix major rendiment que en altres servidors en què només es pot utilitzar PHP com a CGI.

---

4 La web del projecte PHP inclou un apartat que recull diversos estudis d'ús del llenguatge (<http://www.php.net/usage.php>) des de diverses perspectives.

## **5.2. Aspectes de seguretat**

La seguretat d'aquest sistema està centralitzada per disseny. Seguint el patró disseny controlador-façana, hem implementat un sol objecte que tracta totes les peticions que arriben a la aplicació.

Aquest objecte comprova l'autenticació de l'usuari i si aquest té els permisos adequats per realitzar l'acció que demana. Aquesta estructura també ens permet filtrar els paràmetres en un sol punt i evitar atacs d'injecció d'SQL.

### **5.2.1. Sistema d'autenticació**

Hem implementat l'autenticació usant un component que abstregui el sistema, del servidor d'autenticació. Per a aquest objectiu hem escollit el paquet Auth de les llibreries PEAR. Per escollir aquest paquet, hem tingut en compte que permet diversos servidors d'autenticació. Això és molt útil per als usuaris que, amb un sol nom d'usuari i contrasenya, poden accedir a diversos sistemes.

Per a la implementació del nostre sistema, hem usat autenticació per base de dades usant la pròpia base de dades del sistema (per això hi ha creades les taules Usuari i Tipus usuari). En aquest cas el paquet Auth usa el paquet DB (també de les llibreries PEAR) per aconseguir la independència del nostre sistema.

## **5.3. Disseny de la interfície amb l'usuari**

Per dissenyar l'esquema de d'interfície amb l'usuari, primer de tot vam pensar en usar-ne un de molt comú en moltes pàgines web i intranets. En aquest esquema, la pantalla es divideix en tres zones (*Figura 5.1*):

(1) **Barra superior:** On hi ha el títol i algunes capçaleres de la pàgina web. Normalment aquesta barra és fixa i el seu contingut no canvia durant l'ús de l'aplicació.

(2) **Barra lateral:** On hi ha un o més menús amb les accions que es poden dur a terme en relació a l'activitat que s'està fent en cada moment. Normalment, aquesta barra és fixa, però el seu contingut canvia depenent de l'activitat que s'estigui fent.

(3) **Cos:** On hi ha el contingut de la pàgina actual i canvia depenent de les accions que s'escullin a la barra lateral.

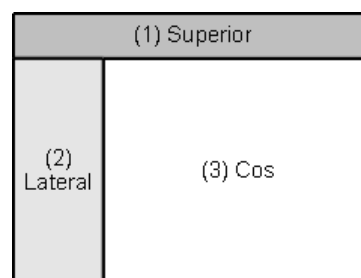


Figura 5.1: Interfície típica

Però aquest esquema típic té l'inconvenient de que redueix la zona de continguts tant en alçada com en amplada. Això s'accentua encara més si la **Barra superior (1)** i la **Barra lateral (2)** són fixes encara que es faci *scroll* pel **Cos (3)** de la pantalla.

Nosaltres necessitem un **Cos (3)** ben espaiós, sobretot per visualitzar la distribució de PCs als laboratoris. Per tant vam prendre una decisió que fa que el disseny sigui menys vistós: continuariem dividint pantalla en 3 zones, però una sobre l'altra (Figura 5.2). A més si fem un *scroll*, totes les tres zones actuen com un bloc i es desplacen totes amunt i avall. Mirem amb més detall cadascuna de les tres zones:

[1] **Barra de capçaleres:** Aquesta barra correspon a la antiga **Barra superior (1)** i hi continua havent el títol i algunes capçaleres de la pàgina web. Concretament hi ha el títol del projecte (que és el mateix en totes les pantalles de l'aplicació) i tres capçaleres:

- La primera capçalera és una barra que anomenem **Barra de Validació** on hi ha els *links* per validar-se o desconnectar-se. En cas d'estar validats, a la **Barra de validació**, també hi apareix el nom d'usuari i un *link* per canviar de *Manteniment de laboratoris* a *Gestió d'imatges* i viceversa.
- La segona capçalera és una altra barra que anomenem **Barra de retorn**. En aquesta barra, hi ha els *links* per tornar als apartats anteriors, d'on procedim.
- I la tercera capçalera és el títol de la pàgina actual dintre de l'aplicació.

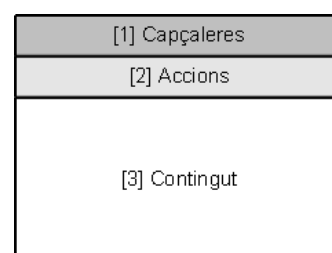


Figura 5.2: Interfície adaptada

A continuació, un exemple de la **Barra de capçaleres [1]**:

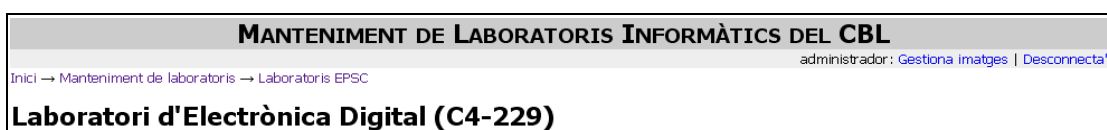


Figura 5.3: Barra de capçaleres [1]



[2] **Barra d'accions:** Aquesta barra correspon a la antiga **Barra lateral (2)** i hi continua havent les accions que es poden dur a treme en relació a l'activitat que s'està fent en cada moment. El seu contingut canvia depenent de l'activitat que s'estigui fent.

En aquesta barra, s'hi poden diferenciar clarament dos tipus d'accions:

- Les primeres, són accions de selecció: ajuden a seleccionar més fàcilment laboratoris, PCs o imatges (depenent de l'apartat de l'aplicació en el que estiguem) de la zona del **Contingut [3]**, podent-los seleccionar tots alhora, cap i bé invertir la selecció.
- El segon tipus d'accions són les que corresponen als casos d'ús del capítol d'Especificació. Aquestes accions es realitzen sobre els elements seleccionats de la zona del **Contingut [3]**.

Aquesta **Barra d'accions** de vegades resulta poc vistosa perquè hi ha diversos botons i menús desplegable alhora. Però això passa perquè, de nou, vam renunciar a un disseny agradable per tenir una aplicació més àgil i funcional. Per exemple, si estem veient la llista d'imatges que tenim, per crear una nova imatge hi ha el botó [Crea Imatge]. Potser es considera normal que, al prémer aquest botó, aparegui una nova pantalla amb un formulari demanant les dades de la nova imatge i, al acceptar-les, es torni a la pantalla on es veu la llista de les imatges, amb la nova imatge inclosa. Doncs a la nostra aplicació, quan es mostren les imatges, també es mostren els camps en blans per crear-ne una de nova. Així, podem omplir directament els camps del formulari i només prement 1 sol cop el botó [Crea Imatge], ja s'afegeix la imatge i es torna a la pantalla on es mostren totes les imatges.

Passa el mateix quan es vol canviar l'estat de PCs o laboratoris, assignar imatges a PCs, o bé fixar el moment d'inici d'una restauració de PCs: tenim a la vista tots els formularis per introduir dades i, un cop introduïdes, només prement un sol botó ja s'executa l'acció.

[3] **Contingut:** Aquesta zona correspon a la antic **Cos (3)**, també hi ha el contingut de la pàgina actual i canvia depenent de les accions que s'escullin a la **Barra d'accions [2]**, o bé a la **Barra de capçaleres [1]**.

En aquesta zona és on es pinta la llista d'edificis, de laboratoris o d'imatges. O també s'hi dibuixa la distribució de PCs en un laboratori o les dades del PC seleccionat. En totes aquestes llistes o distribució d'elements, hi ha un camp *checkbox*, per seleccionar un o més elements de la pantalla i executar en un sol pas, una acció de la **Barra d'accions [2]**.

### 5.4. Documentació

Per escriure aquesta memòria hem usat la *suite* OpenOffice<sup>5</sup> v2.0 que és *software* lliure i que també inclou una eina per convertir els documents de text de l'OpenOffice (\*.odt) a format PDF.

Per fer les figures de la memòria hem usat l'editor de diagrames Dia<sup>6</sup> v0.96 de la fundació Free Software Foundation.

### 5.5. Fitxers de l'aplicació

L'aplicació web consta d'un conjunt de fitxers que descriurem tot seguit. Primer, però m'agradaria dir que tots els fitxers que s'han d'accedir des de web estan dintre del directori ~/public\_html/ que té permisos de lectura i d'execució per a tothom (755). El servidor Apache està configurat per accedir en aquest directori quan es vagi a l'URL del projecte.

- /crea.sql *Script* on hi ha tot el codi SQL: des de la creació de la base de dades i les taules, fins als INSERTS necessaris per poder executar l'aplicació prototipus simulant un ús real.
- /public\_html/index.html Aquest fitxer implementa el ControladorFaçana del patró MVC.
- /public\_html/Controlador.php Aquest fitxer implementa la classe Controlador del patró MVC.
- /public\_html/Vista.php Aquest fitxer implementa la classe Vista del patró MVC.
- /public\_html/Model.php Aquest fitxer implementa la classe Model del patró MVC.
- /public\_html/Control\_Auth.php Implementa la classe Control\_Auth que és la que conté totes les crides a la classe Auth del paquet Auth de les llibreries PEAR.

---

5 Per a més sobre l'OpenOffice, podeu consultar la pàgina web <http://www.openoffice.org/>.

6 Per a més sobre l'editor de diagrames Dia, podeu consultar la pàgina web <http://www.gnome.org/projects/dia/>.

- `/public_html/Control_*.php` Aquests fitxers implementen les classes amb el mateix nom (`Control_*`) i fan tasques corresponents a la lògica del sistema.
- `/public_html/Vista_*.php` Aquests fitxers implementen les classes amb noms `Vista_*` i fan tasques d'interfície entre l'usuari i el sistema.
- `/public_html/Model_*.php` Implementen les classes amb noms `Model_*` i executen totes les consultes i altres comandes a la classe `AccesBD`.
- `/public_html/AccesBD.php` Implementa la classe `AccesBD` i és l'únic que es comunica amb l'SGBD.
- `/public_html/Crea_totes_instancies.php` En aquest projecte hem centralitzat en una classe la creació de totes les instàncies a objectes perquè quedi més ordenat i estructurat. Cada vegada que es vol crear una instància es crida a aquesta classe en comptes de fer un `new` directament.
- `/public_html/estil.css` Aquest fitxer és el que conté tots els fulls d'estil en cascada (CSS) que s'usen a l'aplicació web.
- `/public_html/javascript.js` Aquest fitxer conté tot el codi Javascript que s'usa a la nostra aplicació web.
- `/public_html/etc_dhcp/dhcpd.conf` Aquest fitxer és el que conté la configuració del DHCP. Es llegeix i s'escriu des de la nostra aplicació, però també l'usa el servei de DHCP (degudament configurat). El propietari d'aquest fitxer és l'usuari 'apache'.

### 5.6. Base de dades

#### 5.6.1. Creació de la base de dades

Per crear la base de dades usem part de l'*script* `crea.sql`. Concretament, l'usuari `root` crea la base de dades `ProjecteFIB` i l'usuari `rai` que tindrà permisos totals per a aquesta base de dades. Les sentències SQL són les següents:

```
##Com a 'root'@'localhost'
CREATE USER 'rai'@'localhost' IDENTIFIED BY 'projecte';
GRANT CREATE ON *.* TO 'rai'@'localhost' IDENTIFIED BY 'projecte' WITH
    MAX_QUERIES_PER_HOUR 0 MAX_CONNECTIONS_PER_HOUR 0 MAX_UPDATES_PER_HOUR 0
    MAX_USER_CONNECTIONS 0;
CREATE DATABASE ProjecteFIB DEFAULT CHARACTER SET utf8 COLLATE utf8_spanish_ci;
GRANT ALL PRIVILEGES ON ProjecteFIB.* TO 'rai'@'localhost' IDENTIFIED BY 'projecte'
    WITH GRANT OPTION;
```

#### 5.6.2. Disseny físic de la base de dades

Per crear l'estructura de la base de dades usem una altra vegada l'*script* `crea.sql`. Concretament, l'usuari `rai` crea totes les taules de la base de dades. Les sentències SQL són les següents:

```
##--Com a 'rai'@'localhost'

USE ProjecteFIB;

#####
## CREATes taules ##
#####

CREATE TABLE tbl_tipus_usuari
(
    id_tipus    SMALLINT(5) UNSIGNED NOT NULL AUTO_INCREMENT,
    nom_tipus   VARCHAR(60) NOT NULL,
    PRIMARY KEY (id_tipus)
) TYPE = innodb;
```

```

CREATE TABLE tbl_usuari
(
  nom_usuari  VARCHAR(25) DEFAULT NULL,
  contrasenya VARCHAR(60) DEFAULT NULL,
  tipus       SMALLINT(5) UNSIGNED NOT NULL,
  PRIMARY KEY (nom_usuari),
  FOREIGN KEY fk_tipus (tipus) REFERENCES tbl_tipus_usuari (id_tipus)
    ON DELETE RESTRICT
    ON UPDATE CASCADE
) TYPE = innodb;

CREATE TABLE tbl_ed
(
  numero      VARCHAR (5) NOT NULL,
  nom          VARCHAR(10) DEFAULT NULL,
  nom_llarg    VARCHAR(60) DEFAULT NULL,
  num_labs     SMALLINT(5) UNSIGNED NOT NULL DEFAULT '0',
  PRIMARY KEY (numero)
) TYPE = innodb;

CREATE TABLE tbl_lab
(
  id_lab      VARCHAR(10) NOT NULL,
  numero      VARCHAR (5) NOT NULL,
  num_ed      VARCHAR (5) NOT NULL,
  nom_curt    VARCHAR(10) DEFAULT NULL,
  nom_llarg   VARCHAR(50) DEFAULT NULL,
  num_llocs_treb SMALLINT(5) UNSIGNED NOT NULL DEFAULT '12',
  num_LTs_extra SMALLINT(5) UNSIGNED NOT NULL DEFAULT '0',
  LTs_x_fila  SMALLINT(5) UNSIGNED NOT NULL DEFAULT '3',
  numportes  SMALLINT(5) UNSIGNED NOT NULL DEFAULT '2',
  porta_inici SMALLINT(5) UNSIGNED NOT NULL DEFAULT '1',
  PRIMARY KEY (id_lab),
  FOREIGN KEY fk_num_ed (num_ed) REFERENCES tbl_ed (numero)
    ON DELETE RESTRICT
    ON UPDATE CASCADE
) TYPE = innodb;

CREATE TABLE tbl_PC
(
  nom          VARCHAR(30) NOT NULL,
  ip           VARCHAR(15) NOT NULL,
  mac          VARCHAR(17) NOT NULL,
  sn           VARCHAR(50) DEFAULT NULL,
  hw_num       VARCHAR(50) DEFAULT NULL,
  tipus        VARCHAR(10) DEFAULT NULL,
  lab_pc       VARCHAR(10) NOT NULL,
  LT_lab       SMALLINT(5) NOT NULL DEFAULT '0',
  PRIMARY KEY (nom),
  FOREIGN KEY fk_lab_pc (lab_pc) REFERENCES tbl_lab (id_lab)
    ON DELETE RESTRICT
    ON UPDATE CASCADE
) TYPE = innodb;

CREATE TABLE tbl_relacio_PC_img
(
  pc          VARCHAR(30) NOT NULL,
  img         SMALLINT(5) UNSIGNED NOT NULL,
  PRIMARY KEY (pc, img)
) TYPE = innodb;

CREATE TABLE tbl_tipus_SO
(
  id_so       SMALLINT(5) UNSIGNED NOT NULL AUTO_INCREMENT,
  nom_so      VARCHAR(20) NOT NULL,
  PRIMARY KEY (id_so)
) TYPE = innodb;

```

## 5. Implementació

---

```
CREATE TABLE tbl_imatge
(
  id_imatge      SMALLINT (5) UNSIGNED NOT NULL AUTO_INCREMENT,
  nom_imatge     VARCHAR (20) NOT NULL,
  data_creacio   DATETIME NOT NULL,
  data_ult_act   DATETIME NOT NULL,
  fitxer         VARCHAR(256) NOT NULL,
  so             SMALLINT (5) UNSIGNED DEFAULT NULL,
  PRIMARY KEY    (id_imatge),
  FOREIGN KEY    fk_so (so) REFERENCES tbl_tipus_SO (id_so)
    ON DELETE    RESTRICT
    ON UPDATE    CASCADE
) TYPE = innod;

CREATE TABLE tbl_tipus_estat_lab
(
  id_estat_lab   SMALLINT(5) NOT NULL,
  nom_estat_lab  VARCHAR(30) NOT NULL,
  PRIMARY KEY    (id_estat_lab)
) TYPE=innodb;

CREATE TABLE tbl_tipus_estat_PC
(
  id_estat_pc    SMALLINT(5) NOT NULL,
  nom_estat_pc   VARCHAR(50) NOT NULL,
  PRIMARY KEY    (id_estat_pc)
) TYPE = innodb;

CREATE TABLE tbl_estat_lab
(
  id_estat       SMALLINT(5) UNSIGNED NOT NULL AUTO_INCREMENT,
  lab            VARCHAR(10) NOT NULL,
  instant        DATETIME     NOT NULL,
  tipus_estat    SMALLINT(5) NOT NULL,
  PRIMARY KEY    (id_estat),
  FOREIGN KEY    fk_lab (lab) REFERENCES tbl_lab (id_lab)
    ON DELETE    RESTRICT
    ON UPDATE    CASCADE,
  FOREIGN KEY    fk_tipus_estat (tipus_estat) REFERENCES tbl_tipus_estat_lab (id_estat_lab)
    ON DELETE    RESTRICT
    ON UPDATE    CASCADE
) TYPE = innodb;

CREATE TABLE tbl_estat_PC
(
  id_estat       SMALLINT(5) UNSIGNED NOT NULL AUTO_INCREMENT,
  pc             VARCHAR(30) CHARACTER SET utf8 COLLATE utf8_spanish_ci NOT NULL,
  imatge         SMALLINT(5) UNSIGNED DEFAULT NULL,
  instant        DATETIME     NOT NULL,
  tipus_estat    SMALLINT(5) NOT NULL,
  PRIMARY KEY    (id_estat),
  FOREIGN KEY    fk_pc (pc) REFERENCES tbl_PC (nom)
    ON DELETE    RESTRICT
    ON UPDATE    CASCADE,
  FOREIGN KEY    fk_imatge (imatge) REFERENCES tbl_imatge (id_imatge)
    ON DELETE    RESTRICT
    ON UPDATE    CASCADE,
  FOREIGN KEY    fk_tipus_estat (tipus_estat) REFERENCES tbl_tipus_estat_PC (id_estat_pc)
    ON DELETE    RESTRICT
    ON UPDATE    CASCADE
) TYPE = innodb;
```

### 5.6.3. Primeres insercions

El primer que fem una vegada tenim l'estructura creada, és inserir un parell d'usuaris perquè puguin validar-se:

```
##--Com a 'rai'@'localhost'

USE ProjecteFIB;

#####
## INSERTs Tipus usuari ##
#####

INSERT INTO tbl_tipus_usuari (id_tipus, nom_tipus)
VALUES ('1', 'Administradors');

#####
## INSERTs Usuaris ##
#####

INSERT INTO tbl_usuari (nom_usuari, contrasenya, tipus)
VALUES ('raimon', MD5('projecte'), '1');
INSERT INTO tbl_usuari VALUES ('administrador', MD5('admin'), '1');
```

A partir d'aquí inserim els edificis, laboratoris, PCs, imatges, SOs, estats, etcètera per tenir la base de dades preparada per usar aquest prototip.

## 5.7. Exemples de codi

### 5.7.1. index.html: el controlador façana

Incloem aquest fitxer perquè creiem que és una manera prou entenedora de veure què fa un controlador façana:

```
<?php
...

//Creem els objectes de les classes Control_Auth i Controlador
$sc_auth = Crea_totes_instancias::CreaInstC_Auth();
$sccontrol = Crea_totes_instancias::CreaInstControlador();
```

## 5. Implementació

```
switch ($_REQUEST["accio"])
{
    case "Login":
        //Validació:          Valida
        $control->Valida($c_auth);
        break;

    case "Logout":
        //Validació:          Desconnecta
        $control->Desconnecta($c_auth);
        break;

    case "Llista_eds":
        //Consulta a l'inventari:      Mostra edificis
        $control->MostraEds($c_auth);
        break;

    case "Mostra_ed":
        $ed = $_REQUEST["num_ed"];

        //Manteniment de laboratoris: Restaura laboratoris      i
        //Gestió d'estats:      Modifica estat laboratoris
        $control->AccionsEd($ed, $c_auth);

        //Consulta a l'inventari:      Mostra laboratoris
        $control->MostraLabs_perEd($ed,$c_auth);
        break;

    case "Mostra_lab":
        $id_lab = $_REQUEST["id_lab"];

        //Manteniment de PCs:      Restaura PCs      i
        //Gestió d'estats:      Canvia estat PCs
        $control->AccionsLab($id_lab, $c_auth);

        //Consulta a l'inventari:      Mostra PCs
        $control->MostraPCs_perLab($id_lab, $c_auth);
        break;

    //Gestió d'imatges
    case "Gestiona_imgs":
        $mod_o_esb = "";
        $id_img_x_mod_esb = "";

        switch ($_REQUEST["botoAccionsImgs"])
        {
            case "CreaImg":
                //Gestió d'imatges:      Crea imatge
                $control->CreaImatge($c_auth);
                break;
            case "ModifImg":
                $mod_o_esb = "MOD";
                $id_img_x_mod_esb = $_REQUEST["img_seleccionada"];
                break;
            case "EsborrImg":
                $mod_o_esb = "ESB";
                $id_img_x_mod_esb = $_REQUEST["img_seleccionada"];
                break;
            default:
                switch ($_REQUEST["botoConfirma"])
                {
                    case "SiMod":
                        //Gestió d'imatges:      Modifica imatge
                        $control->ModificaImatge($c_auth);
                        break;
                    case "SiEsb":
                        //Gestió d'imatges:      Esborra imatge
                        $control->EsborraImatge($c_auth);
                        break;
                }; //Fi 3er switch Gestiona_imgs
        }; //Fi 2on switch Gestiona_imgs

        $control->GestionaImatges($mod_o_esb, $id_img_x_mod_esb, $c_auth);
        break;

    default:
        $control->MostraEds($c_auth);
}; //Fi 1er switch

?>
```



## 5.8. Proves

A mida que el projecte anava avançant, ja l'hem anat provant. Concretament, hi ha hagut parts de la implementació que hem fet en bloc i que coincidien amb algun requeriment funcional. Per exemple, la part de validació o la part de gestió d'imatges, les hem implementat i tot seguit les hem provades bastant exhaustivament. Per tant, aquest parell de requeriments funcionals, ja es podien considerar provats.

Així, poc a poc, hem anat provant que es complissin tots els altres requisits funcionals descrits al capítol d'Anàlisi de requisits i tots els casos d'ús del capítol d'especificació.

A més, hem realitzat diverses insercions a la base de dades de tal manera d'intentar provar totes les possibilitats que hi pugui haver a la realitat inclosa la creació de diversos elements amb noms i paràmetres erronis i amb caràcters estranys, o bé buidar algunes taules imprescindibles per veure com reacciona l'aplicació.

Hem provat de restaurar una i més imatges en diversos ordinadors de *hardware* diferent.

Hem provat de fer d'usuari malintencionat per intentar accedir a zones de l'aplicació sense permís.

Hem provat d'accedir a la aplicació simultàniament des de diversos llocs.

El resultat d'aquestes proves han permès detectar i corregir diversos errors.



## 6. Conclusions

6.1. Balanç.....	105
6.2. Línies de futur.....	106

### 6.1. Balanç

Arribats a aquest punt, és moment de fer balanç: hem acomplert l'objectiu general de crear un prototipus d'un sistema accessible via web capaç de restaurar remotament les imatges dels PCs dels diversos laboratoris del CBL.

El nostre sistema també ha assolit la resta d'objectius específics enumerats al primer capítol d'aquesta memòria. Hem implementat un entorn adaptat als diversos tipus d'usuaris i hem aconseguit una interfície web intuïtiva i funcional. També hem creat una base de dades amb la possibilitat d'incloure-hi algunes dades que, actualment, no estan ben inventariades. I, a més, accedint a aquesta base de dades via web usant el nostre sistema, podrem saber en temps real quin és l'estat dels laboratoris informàtics del CBL.

El sistema implementat té una portabilitat acceptable, que permetria integrar-lo en altres campus retocant-ne ben poca cosa. A més, tot i que el sistema només es pot connectar a bases de dades MySQL, modificant tan sols l'arxiu AccesBD, es pot adaptar fàcilment a tot tipus de bases de dades, per exemple usant PEAR DB per fer-ho.

A nivell personal, el desenvolupament d'aquest projecte m'ha comportat un esforç molt més gran del que es pot apreciar a simple vista: ja fa força anys vaig fer les assignatures de programació de la carrera i el lloc on estic treballant no em requereix programar. Per causa d'això, he de confessar que mai havia usat la programació orientada a objectes, ni havia fet mai cap web dinàmica. A més, com que he cursat la carrera d'Enginyeria Tècnica en Informàtica de Sistemes, no havia fet cap assignatura de l'enginyeria del *software*.

És per això que, tot i haver-me hagut d'esforçar una mica més que si ja tingués experiència en aquest tipus d'aplicacions, el desenvolupament d'aquest projecte m'ha comportat una gran satisfacció personal perquè m'he posat més o menys al dia en molts temes en els que estava desfasat: he tingut un primer contacte amb el disseny d'aplicacions més o menys grans, he comprès molt millor què és i perquè serveix l'UML, he practicat per primera vegada el disseny de webs dinàmiques (descobrint el PHP) i els fulls d'estil en cascada (CSS).

I també he pogut posar en pràctica alguns coneixements adquirits durant la carrera com en l'àmbit de les bases de dades i les aplicacions en entorns distribuïts.

### 6.2. Línies de futur

És fàcil marcar unes línies de futur per ampliar aquest projecte. De seguida en veiem una de claríssima:

La primera línia de futur que se'ns acudeix és que aquest sistema, que és un sistema prototipus, es pugui posar en explotació al CBL. Realment, perquè això es pogués dur a terme s'haurien de fer molt poques modificacions. Inicialment només caldria introduir nous usuaris a la base de dades i també entrar-hi totes les dades de tots els PCs i laboratoris del campus. Llavors provaríem com respon el sistema amb una càrrega real. Després, es faria una avaluació pels caps de l'Àrea Tècnica del CBL que posposarien algunes modificacions per poder-lo posar en explotació.

Una segona possible ampliació començaria notant que tot aquest projecte l'hem desenvolupat usant *software* lliure, excepte la part de restauració d'imatges que l'hem feta amb *software* comercial: el Rembo. La millora consistiria a reimplementar el sistema usant només *software* lliure. La versió antiga del Rembo, que es deia BpBatch, era de codi obert i ja implementava un programa que es podia enviar per TFTP al ordinadors clients perquè aquests l'executessin en arrencar. Podem trobar un *mini-HOWTO* de com funcionava i descarregar-nos-el a la següent URL: <http://tldp.org/HOWTO/Remote-Boot.html>.

El que podríem fer llavors, seria investigar sobre aquest NBP (*Network Boot Program*) perquè el nostre sistema l'enviés i podríem executar codi als ordinadors client.

Però encara ens faltaria trobar un programa lliure que ens generés i restaurés imatges. Ja n'hem vist un a l'apartat de Treballs relacionats del capítol 2. *Descripció de l'entorn del sistema* que es deia DriveImage (<http://www.drive-image.com/>), però aquest potser no es pot executar en *silent-mode*. El que sí que segurament aniria bé seria el programa Ghost for Linux<sup>7</sup> (G4L) que prové d'un projecte de Sourceforge: <http://sourceforge.net/projects/g4l>.

I no tindríem cap problema per pensar en més línies de futur per aquest projecte, com per exemple ampliar la part d'inventari del nostre sistema i fer que poguéssim afegir, modificar i esborrar dades des de la pròpia aplicació. Podríem fer el mateix per als usuaris (i les seves contrasenyes).

A més, una vegada estiguessin fetes les modificacions del paràgraf anterior, Podríem usar aquest inventari de PCs, laboratoris edificis i imatges com a inventari principal de recursos informàtics del CBL, que actualment està en Lotus Notes.

---

7 Per a més informació sobre el Ghost for Linux, també podem consultar aquesta URL: <http://www.csc.liv.ac.uk/~greg/ghost4linux/index.html>.

## 7. Bibliografia

### Llibres

- [1] Tim Converse and Joyce Park with Clark Morgan. *PHP5 and MySQL® Bible*. Wiley Publishing 2004.
- [2] David Sklar, Adam Trachtenberg. *PHP Cookbook*. O'Reilly November 2002.
- [3] Steve Suehring. *The MySQL Bible*. Wiley. 2003.
- [4] Dolors Costal, M. Ribera Sancho, Ernest Teniente. *Enginyeria del software. Especificació. Especificació de sistemes orientats a objectes amb la notació UML*. Edicions UPC 2000.
- [5] Cristina Gómez Seoane, Enric Mayol Sarroca, Antoni Olivé Ramon, Ernest Teniente López. *Enginyeria del software: Disseny I. Disseny de sistemes orientats a objectes usant la notació UML*. Edicions UPC 2003.

### Articles

- [6] Alberto Saviola. *Web Page Response time 101*.
- [7] José María Morales Vázquez. *Una introducción a las pruebas de prestaciones*.

### Pàgines web

- [8] World Wide Web Consortium: <http://www.w3.org/>
- [9] Manual de PHP: <http://www.php.net/manual/en/index.php>
- [10] Manual de PEAR Auth: <http://pear.php.net/package/Auth/>
- [11] MySQL 5.0 Reference Manual: <http://dev.mysql.com/doc/refman/5.0/es/index.html>
- [12] Practical UML: A Hands-On Introduction: <http://dn.codegear.com/article/31863/>
- [13] Unified Modeling Language: <http://www-306.ibm.com/software/rational/uml/>
- [14] Manual de CSS en català: <http://dense13.com/wec/curscss/>
- [15] JavaScript Tutorial: <http://www.w3schools.com/js/default.asp>